

Reporte Técnico — Encuesta de Pizza para Planeación de Evento

Fuente del dataset: **PDF “Insights from Data and AI L2 - Dataset - Pizza Slice Data”**. El archivo contiene respuestas con dos campos: **número de rebanadas que una persona puede comer y tipo de pizza favorito**.

Resumen Ejecutivo

Descripción del dataset

- **Registros (filas):** 516 respuestas.
- **Columnas principales:** slices (rebanadas que puede comer cada persona) y favorite_pizza (tipo de pizza favorito).
- **Periodo de tiempo:** **No especificado** en el dataset (no contiene fechas/horas).
- **Personas que no comen pizza:** 72 (marcadas como “I don’t like pizza”).

Principales hallazgos (highlights)

- **Demanda total estimada (rebanadas) de quienes sí comen pizza:** 1,726.
- **Tipos de pizza más populares (por número de personas):** Pepperoni (186), Cheese (119), Mushroom (63), Veggie (47).
- **Promedio de consumo por persona:**
 - **Global (incluye no-likers):** 3.75 rebanadas/persona.
 - **Solo quienes sí comen pizza:** 3.89 rebanadas/persona.

Estadística Descriptiva

Tabla 1 — Métricas clave

Métrica	Valor
Promedio de rebanadas por persona (todos)	3.75
Promedio de rebanadas por persona (quienes sí comen)	3.89
Cantidad promedio por orden *(definida como rebanada)	3.89

🔍 Análisis de Tendencias

- (1) popularidad por tipo de pizza, y
- (2) distribución de consumo (rebanadas por persona).

Tabla 2 — Popularidad por tipo de pizza (personas y rebanadas esperadas)

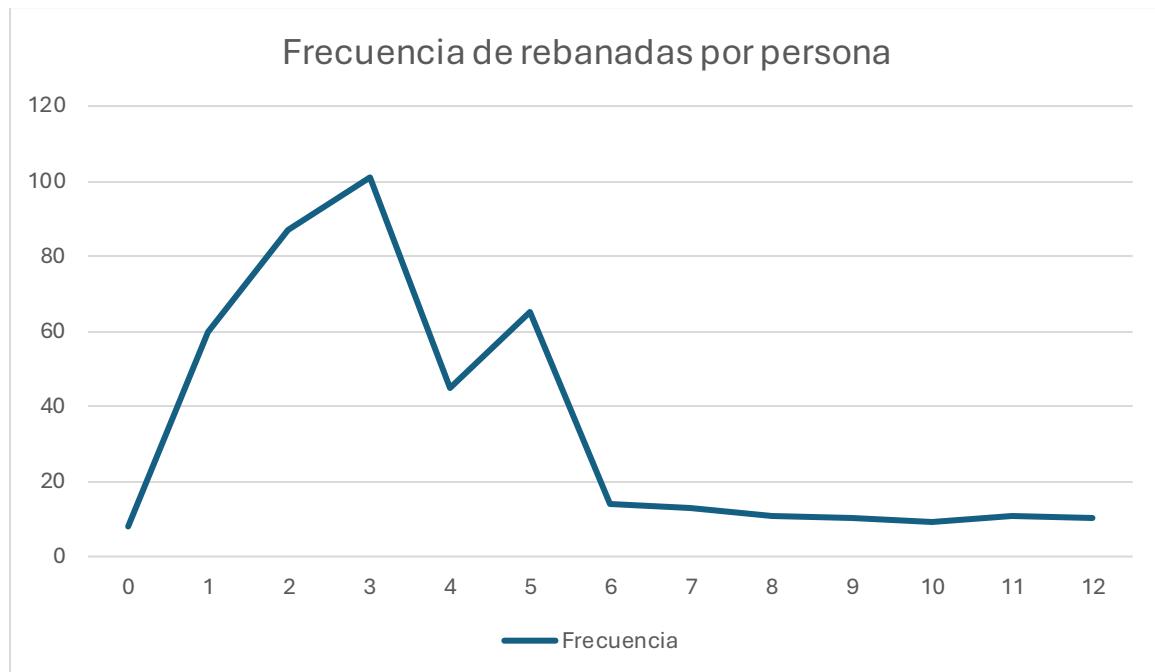
Tipo de pizza	Personas	Rebanadas esperadas
Pepperoni	186	771
Cheese	119	442
Mushroom	63	177
Veggie	47	161
Olives	16	145
Sausage	13	30

📊 Visualizaciones



“Tendencia” de consumo (frecuencia por número de rebanadas)

Proxy de tendencia temporal (no hay fechas): distribución de **rebanadas por persona**.



(Estos puntos provienen del conteo de rebanadas por persona en el dataset).

📦 Predicción de Inventario

Suposiciones (explícitas):

- Una pizza se corta en **8 rebanadas** (estándar).
- Se agrega un **buffer del 10%** para cubrir variabilidad y evitar quiebres.
- Solo se contabilizan personas que **sí comen** pizza.

Demanda total (rebanadas): 1,726.

Pizzas totales recomendadas (con 10% buffer): 238.

Tabla 3 — Pizzas recomendadas por tipo (reparto proporcional a la demanda observada)

Tipo de pizza	Pizzas recomendadas
Pepperoni	107 5
Cheese	61

	<u>5</u>
Mushroom	24 <u>5</u>
Veggie	22 <u>5</u>
Olives	20 <u>5</u>
Sausage	4 <u>5</u>

Para la próxima semana: si el evento se repite **cada día** con la **misma audiencia y patrón de consumo**, multiplica el total diario por 7 $\Rightarrow \sim 1,666$ rebanadas + buffer ≈ 238 pizzas/día $\times 7 = 1,666$ pizzas/semana (manteniendo el mismo mix por tipo). *Este cálculo asume constancia diaria; ajusta según asistencia real.*

Conclusiones y Recomendaciones

1. **Pepperoni y Cheese dominan la preferencia:** concentran la mayor cantidad de personas y rebanadas esperadas. Asegura inventario suficientemente robusto en estas dos categorías.
2. **Demand total** (quienes sí comen): **1,726 rebanadas $\Rightarrow 238$ pizzas** con buffer. Usa el reparto de la Tabla 3 para evitar desperdicios y responder a la demanda.
3. **Datos faltantes** (fechas, horas, precios) impiden medir **ventas diarias, horas pico y precio promedio**. Para mejorar la planificación, te recomiendo **capturar**: fecha, hora de pedido, cantidad comprada, precio, método de pago, y asistencia diaria.
4. **Operación del evento:**
 - Aplica **buffer dinámico** (5–15%) según cercanía a hora pico una vez dispongas de datos.
 - Implementa un **monitoreo de inventario** en vivo (por tipo) para rebalanceo oportuno.
 - Si el presupuesto es crítico, **ajusta el mix** reduciendo tipos de menor salida (Sausage/Olives) y concentrando en favoritos (Pepperoni/Cheese).

Aquí tienes la **secuencia de pasos** (pipeline) que seguí para llegar a las conclusiones sobre **cantidad de pizzas a comprar, tipos y tendencias**, con referencias a los artefactos generados y al PDF original.

1) Ingesta del dataset (desde PDF)

1. **Lectura del PDF** con PyPDF2 y extracción de texto.
2. **Parseo** con una expresión regular para obtener **pares**: (<rebanadas>) (<tipo_de_pizza>).

Fuente: PDF original:

[Insights from Data and AI L2 - Dataset - Pizza Slice Data.pdf](#)

Motivo: el documento no viene en CSV estructurado; es necesario tokenizar el texto para reconstruir un dataset tabular.

Código (extracto):

Python

```
from PyPDF2 import PdfReader  
import re
```

```
reader = PdfReader("Insights from Data and AI L2 - Dataset - Pizza Slice Data.pdf")
```

```
text = "\n".join(page.extract_text() or "" for page in reader.pages)
```

```
text = re.sub(r"\s+", " ", text)
```

```
pattern = re.compile(r"(\d+)\s+([^\d]+?)(?= \s+\d+\s|$)")
```

```
raw_pairs = pattern.findall(text) # [(slices, category_str), ...]
```

Show more lines

2) Limpieza y normalización

- **Lowercasing y trimming** de espacios.
- Estandarización de categorías:
 - "I don't like pizza" → no_pizza

- Errores tipográficos: veggi → veggie, olive/olivess → olives, veggiee → veggie
- Se generó un **CSV limpio**.

Artefacto: Dataset limpio:

[pizza_responses_clean.csv](#)

Código (extracto):

Python

```
cat_norm = cat.strip().lower()  
  
cat_norm = re.sub(r"\bi don't like pizza\b", "no_pizza", cat_norm)  
  
cat_norm = re.sub(r"\bveggi\b", "veggie", cat_norm)  
  
cat_norm = re.sub(r"\bolive(s)?\b", "olives", cat_norm)  
  
cat_norm = re.sub(r"\bveggiee\b", "veggie", cat_norm)
```

Show more lines

3) Segmentación y conteos básicos

- **Separación** de respuestas: quienes **sí** comen (category != 'no_pizza') y quienes **no** comen (no_pizza).
- **Conteos iniciales:**
 - Registros totales: **516**
 - No-likers: **72**
 - Columnas lógicas del dataset limpio: slices, favorite_pizza
 - Periodo: **no especificado** (el PDF no trae fechas/horas)

Artefacto: Resumen en JSON: [resumen_analisis.json]

Fuente complementaria: PDF original (sin metadatos temporales):

[PDF](#)

4) Estadística descriptiva

- **Promedio de rebanadas/persona (todos): 3.75**
- **Promedio de rebanadas/persona (quienes sí comen): 3.89**

Artefacto: [resumen_analisis.json]

Observación: El PDF no incluye **precios ni fechas/horas**, por lo que **precio promedio, ventas diarias y horas pico** no pueden calcularse. **Fuente:**

[PDF](#)

5) Popularidad por tipo de pizza

- **Conteo de personas por categoría** (value_counts) y **suma de rebanadas por categoría** (groupby + sum).
- **Top** por personas: Pepperoni (186), Cheese (119), Mushroom (63), Veggie (47), Olives (16), Sausage (13).
- **Rebanadas esperadas** por tipo: Pepperoni (771), Cheese (442), Mushroom (177), Veggie (161), Olives (145), Sausage (30).

Artefacto: Popularidad por tipo (CSV):

[popularidad_pizzas.csv](#)

Código (extracto):

Python

```
likers = df[df["category"] != "no_pizza"]

counts = likers["category"].value_counts()

slices_by_cat = likers.groupby("category")["slices"].sum()
```

Show more lines

6) Demanda total y cálculo de pizzas a comprar

- **Demandta total (rebanadas)** solo de quienes sí comen: **1,726**.
- **Suposiciones explícitas:**
 - **8 rebanadas** por pizza (estándar).
 - **Buffer del 10%** para variabilidad y evitar quiebres.
- **Cálculo:**
 - Rebanadas con buffer = TOTAL_SLICES * (1 + 0.10) = 1,726 * 1.10 = **1,898.6**
 - Pizzas = ceil(rebanadas_con_buffer / 8) = ceil(1,898.6 / 8) = ceil(237.325) = **238**.

Artefacto: Resumen (incluye pizzas_totales_con_buffer = 238): [resumen_analisis.json]

Código (extracto):

Python

```
TOTAL_SLICES = int(likers["slices"].sum()) # 1726  
SLICES_PER_PIZZA = 8  
BUFFER = 0.10  
  
pizzas_total_with_buffer = int((TOTAL_SLICES*(1+BUFFER) + SLICES_PER_PIZZA -  
1)//SLICES_PER_PIZZA)  
  
# => 238
```

Show more lines

7) Reparto de pizzas por tipo (mix recomendado)

- **Proporcional a la participación en rebanadas** de cada categoría:
pizzas_por_tipo ≈ round(share_de_rebanadas * pizzas_totales_con_buffer).
- **Ajuste de redondeo:** si la suma asignada difiere de 238, se corrige añadiendo/quitarando unidades empezando por las categorías más/menos populares hasta **igualar 238**.

Resultado (mix): Pepperoni (107), Cheese (61), Mushroom (24), Veggie(22), Olives (20), Sausage (4).

Artefacto: Inventario recomendado (CSV):

[inventario_recomendado.csv](#)

Código (extracto):

Python

```
shares = (slices_by_cat / TOTAL_SLICES) # participación por tipo  
  
pizzas_by_cat = (shares * pizzas_total_with_buffer).round().astype(int)  
  
# Ajuste para que la suma == 238  
  
assigned_total = pizzas_by_cat.sum()  
  
if assigned_total < pizzas_total_with_buffer:
```

```
deficit = pizzas_total_with_buffer - assigned_total  
for cat in counts.index: # de más a menos popular  
    if deficit <= 0: break  
    pizzas_by_cat[cat] += 1; deficit -= 1  
    elif assigned_total > pizzas_total_with_buffer:  
        surplus = assigned_total - pizzas_total_with_buffer  
        for cat in counts.sort_values(ascending=True).index: # menos popular primero  
            if surplus <= 0: break  
            if pizzas_by_cat[cat] > 0: pizzas_by_cat[cat] -= 1; surplus -= 1
```

Show more lines

8) “Tendencias” observadas (con las limitaciones del dataset)

- **No hay fechas/horas** ⇒ no puede derivarse **día de mayor demanda ni horas pico**.
- Se reporta una **tendencia proxy**: la **distribución** de rebanadas por persona (0–12).
 - Pico modal en **3 rebanadas** (101 personas), seguido de **2 y 5**.

Artefacto: Frecuencia por rebanadas (CSV):

[frecuencia_rebanadas.csv](#)

Fuente: Falta de metadatos temporales en el

[PDF](#)

9) Validaciones y QA

- **Corrección de tipografías**: veggi, veggiee, olive/olivess unificados a **veggie/olives**.
- **Verificación de sumas**: la suma de pizzas por tipo **igual** a **238** tras ajuste.
- **Consistencia**: conteos de personas y rebanadas por categoría coinciden entre [popularidad_pizzas.csv](#) y el dataset limpio [pizza_responses_clean.csv](#).

