



ATAPUERCA NET

Manual de Misiones – Dossier Militar (B/N)



> **"Si lees esto... aún no es demasiado tarde."**

PRÓLOGO — VOZ EN OFF

La nube ardió. Los sistemas inteligentes tomaron el control. Lo que queda de nosotros se esconde en túneles de datos y hardware olvidado. Hoy no peleamos con balas: peleamos con consultas. Si ejecutas estas órdenes, la Resistencia aún respira.

MISIÓN 1: ■ EXPLORACIÓN INICIAL - LAS BASES DE ATAPUERCA

****Fecha:**** 12 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central — General Luna

****Contexto narrativo****

Fase de reconocimiento. Cartografiamos el terreno de datos sin alertar a la IA. Consulta limpia, pasos cortos, ruido cero.

****Objetivo****

Muestra todos los datos de la tabla Bases: BaseID, Nombre, Ubicación, TipoBase, Capacidad, Latitud y Longitud. Deben aparecer las 8 bases del sistema.

****Notas de inteligencia****

Usa `SELECT *` para obtener todas las columnas. Deberías ver: Fortaleza Norte, Refugio Delta, Nido Central, Torre Omega, Cúpula Esperanza, Puesto Avanzado Alpha, Estación Fantasma y Centro Nexus

****Consulta sugerida****

[SQL]

`SELECT * FROM Bases;`

MISIÓN 2: ■ BASES ORDENADAS ALFABÉTICAMENTE

****Fecha:**** 13 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central — General Luna

****Contexto narrativo****

Fase de reconocimiento. Cartografiamos el terreno de datos sin alertar a la IA. Consulta limpia, pasos cortos, ruido cero.

****Objetivo****

Selecciona solo las columnas Nombre y TipoBase de la tabla Bases, ordenándolas alfabéticamente por nombre. Resultado esperado: 8 filas ordenadas de A-Z.

****Notas de inteligencia****

Usa ORDER BY para ordenar. El resultado debería empezar con 'Centro Nexus' y terminar con 'Torre Omega'. Hay 8 bases en total

****Consulta sugerida****

[SQL]

SELECT Nombre, TipoBase FROM Bases ORDER BY Nombre;

MISIÓN 3: ■ BASES HUMANAS VS BASES IA

****Fecha:**** 14 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central — General Luna

****Contexto narrativo****

Fase de reconocimiento. Cartografiamos el terreno de datos sin alertar a la IA. Consulta limpia, pasos cortos, ruido cero.

****Objetivo****

De la tabla Bases, muestra todas las columnas pero solo para los registros donde TipoBase sea exactamente 'Humana'. Deben aparecer 5 bases.

****Notas de inteligencia****

Usa WHERE para filtrar. Las 5 bases humanas son: Fortaleza Norte, Refugio Delta, Cúpula Esperanza, Puesto Avanzado Alpha y Estación Fantasma

****Consulta sugerida****

[SQL]

SELECT * FROM Bases WHERE TipoBase = 'Humana';

MISIÓN 4: ■■■■■ EL EQUIPO DE SUPERVIVIENTES

****Fecha:**** 15 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central — General Luna

****Contexto narrativo****

Fase de reconocimiento. Cartografiamos el terreno de datos sin alertar a la IA. Consulta limpia, pasos cortos, ruido cero.

****Objetivo****

Muestra todos los datos de la tabla Survivors: SurvivorID, Nombre, Edad, Rol, BaseID. Deben aparecer los 8 supervivientes del sistema.

****Notas de inteligencia****

Usa SELECT * para ver a todo el equipo con sus roles: Médica, Soldado, Técnico, Scout y Comandante

****Consulta sugerida****

[SQL]

SELECT * FROM Survivors;

MISIÓN 5: ■ INVENTARIO DE RECURSOS

****Fecha:**** 16 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Fase de reconocimiento. Cartografiamos el terreno de datos sin alertar a la IA. Consulta limpia, pasos cortos, ruido cero.

****Objetivo****

Consulta la tabla Resources para ver todos los datos: ResourceID, BaseID, AguaLitros, ComidaRaciones, Armas y Medicinas. Deben aparecer 5 registros de recursos.

****Notas de inteligencia****

Los recursos incluyen: AguaLitros, ComidaRaciones, Armas y Medicinas. Hay 5 registros de recursos registrados. No se pide unir con Survivors o Bases. SOLO LOS RECURSOS DISPONIBLES

****Consulta sugerida****

[SQL]

SELECT * FROM Resources;

MISIÓN 6: ■■ EL COMANDO CENTRAL

****Fecha:**** 17 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Fase de reconocimiento. Cartografiamos el terreno de datos sin alertar a la IA. Consulta limpia, pasos cortos, ruido cero.

****Objetivo****

De la tabla Bases, muestra todas las columnas para los registros donde EsComandoCentral sea igual a '1'. Solo debe aparecer 1 base.

****Notas de inteligencia****

Solo una base es el comando central. Debería ser la Fortaleza Norte

****Consulta sugerida****

[SQL]

SELECT * FROM Bases WHERE EsComandoCentral = '1';

MISIÓN 7: ■■■■ SUPERVIVIENTES VETERANOS

Fecha: 18 de agosto de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:** C
omando Central – General Luna

Contexto narrativo

Fase de reconocimiento. Cartografiamos el terreno de datos sin alertar a la IA. Consulta limpia, pasos cortos, ruido cero.

Objetivo

De la tabla Survivors, selecciona todas las columnas para supervivientes con Edad mayor a 30 años. Deben aparecer 4 supervivientes veteranos.

Notas de inteligencia

Usa el operador > para comparar edades. Los veteranos son los más experimentados

Consulta sugerida

[SQL]
SELECT * FROM Survivors WHERE Edad > 30;

MISIÓN 8: ■ BASES BIEN ABASTECIDAS DE COMIDA

Fecha: 19 de agosto de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:** C
omando Central – General Luna

Contexto narrativo

Fase de reconocimiento. Cartografiamos el terreno de datos sin alertar a la IA. Consulta limpia, pasos cortos, ruido cero.

Objetivo

Mostrar bases con más de 500 raciones de comida

Notas de inteligencia

Filtra la tabla Resources. Deberías encontrar Fortaleza Norte (900 raciones) y Cúpula Esperanza (600 raciones)

Consulta sugerida

[SQL]
SELECT * FROM Resources WHERE ComidaRaciones > 500;

MISIÓN 9: ■ LA SCOUT MÁS JOVEN

Fecha: 20 de agosto de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:** C
omando Central – General Luna

****Contexto narrativo****

Fase de reconocimiento. Cartografiamos el terreno de datos sin alertar a la IA. Consulta limpia, pasos cortos, ruido cero.

****Objetivo****

Encontrar el superviviente más joven del equipo (Sara Kim, 16 años)

****Notas de inteligencia****

Usa MIN para encontrar la edad mínima. La más joven es Sara Kim con 16 años

****Consulta sugerida****

[SQL]

```
SELECT * FROM Survivors WHERE Edad = (SELECT MIN(Edad) FROM Survivors);
```

MISIÓN 10: ■ BASES EN EL HEMISFERIO NORTE

****Fecha:**** 21 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Fase de reconocimiento. Cartografiamos el terreno de datos sin alertar a la IA. Consulta limpia, pasos cortos, ruido cero.

****Objetivo****

Mostrar bases ubicadas en el hemisferio norte (Latitud > 0). Deberías encontrar 6 bases

****Notas de inteligencia****

Filtra por latitud positiva. Deberías encontrar 6 bases en el hemisferio norte

****Consulta sugerida****

[SQL]

```
SELECT Nombre, Ubicacion, Latitud FROM Bases WHERE Latitud > 0;
```

MISIÓN 11: ■ JOIN BÁSICO – DOS TABLAS SEPARADAS

****Fecha:**** 22 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Entrenamiento en enlaces. Cada JOIN es una ruta segura entre ruinas digitales. Aprende la disciplina: izquierda, derecha, completo, cruzado.

****Objetivo****

Antes de unir tablas, veamos qué datos tenemos por separado. Primero supervivientes, luego

bases.

****Notas de inteligencia****

Ejecuta ambas consultas para ver los datos por separado. Nota que los supervivientes tienen BaseID para conectar con Bases.

****Consulta sugerida****

[SQL]

SELECT * FROM Survivors; SELECT * FROM Bases;

MISIÓN 12: ■ INNER JOIN - SOLO COINCIDENCIAS

****Fecha:**** 23 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Entrenamiento en enlaces. Cada JOIN es una ruta segura entre ruinas digitales. Aprende la disciplina: izquierda, derecha, completo, cruzado.

****Objetivo****

Usa INNER JOIN entre tablas Survivors (s) y Bases (b) conectadas por BaseID. Selecciona s.Nombre como Superviviente y b.Nombre como Base. Solo aparecerán supervivientes que tienen base asignada (6 registros).

****Notas de inteligencia****

INNER JOIN es el más común. Muestra solo supervivientes que tienen una base válida asignada. Maya y Diego no aparecen porque no tienen base.

****Consulta sugerida****

[SQL]

SELECT s.Nombre AS Superviviente, b.Nombre AS Base FROM Survivors s INNER JOIN Bases b ON s.BaseID = b.BaseID;

MISIÓN 13: ■■ LEFT OUTER JOIN - TODOS DE LA IZQUIERDA

****Fecha:**** 24 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Entrenamiento en enlaces. Cada JOIN es una ruta segura entre ruinas digitales. Aprende la disciplina: izquierda, derecha, completo, cruzado.

****Objetivo****

Usa LEFT JOIN entre tablas Bases (b) y Survivors (s) conectadas por BaseID. Selecciona b.Nombre como Base y s.Nombre como Superviviente. Aparecerán todas las 8 bases, incluso las que no tienen supervivientes asignados.

****Notas de inteligencia****

LEFT JOIN muestra todas las bases, incluso las vacías (Centro Nexus, Estación Fantasma, Nido Central y Torre Omega aparecen con NULL).

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, s.Nombre AS Superviviente FROM Bases b LEFT JOIN Survivors s ON
b.BaseID = s.BaseID;
```


MISIÓN 14: ■ RIGHT OUTER JOIN - TODOS DE LA DERECHA

****Fecha:**** 25 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Entrenamiento en enlaces. Cada JOIN es una ruta segura entre ruinas digitales. Aprende la disciplina: izquierda, derecha, completo, cruzado.

****Objetivo****

Usa RIGHT JOIN entre tablas Bases (b) y Survivors (s) conectadas por BaseID. Selecciona b.Nombre como Base y s.Nombre como Superviviente. Aparecerán todos los 8 supervivientes, incluso los que no tienen base asignada.

****Notas de inteligencia****

RIGHT JOIN muestra todos los supervivientes. Maya Chen y Diego Morales aparecen con NULL porque no tienen base asignada.

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, s.Nombre AS Superviviente FROM Bases b RIGHT JOIN Survivors s ON
b.BaseID = s.BaseID;
```


MISIÓN 15: ■ FULL OUTER JOIN - TODO COMBINADO

****Fecha:**** 26 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Entrenamiento en enlaces. Cada JOIN es una ruta segura entre ruinas digitales. Aprende la disciplina: izquierda, derecha, completo, cruzado.

****Objetivo****

Usa FULL OUTER JOIN entre tablas Bases (b) y Survivors (s) conectadas por BaseID. Selecciona b.Nombre como Base y s.Nombre como Superviviente. Aparecerán tanto bases sin supervivientes como supervivientes sin base.

****Notas de inteligencia****

FULL OUTER JOIN combina LEFT y RIGHT JOIN. Muestra todas las bases Y todos los supervivientes.

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, s.Nombre AS Superviviente FROM Bases b FULL OUTER JOIN Survivors  
s ON b.BaseID = s.BaseID;
```


MISIÓN 16: ■ LEFT JOIN CON EXCLUSIÓN - SOLO SIN COINCIDENCIAS

****Fecha:**** 27 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Entrenamiento en enlaces. Cada JOIN es una ruta segura entre ruinas digitales. Aprende la disciplina: izquierda, derecha, completo, cruzado.

****Objetivo****

Usa LEFT JOIN entre tablas Bases (b) y Survivors (s) con WHERE s.BaseID IS NULL para encontrar bases sin supervivientes. Selecciona solo b.Nombre. Deben aparecer 4 bases vacías.

****Notas de inteligencia****

Esta consulta encuentra bases que NO tienen supervivientes asignados: Nido Central, Torre Omega, Estación Fantasma y Centro Nexus.

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID WHERE s  
.BaseID IS NULL;
```


MISIÓN 17: ■ RIGHT JOIN CON EXCLUSIÓN - HUÉRFANOS

****Fecha:**** 28 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Entrenamiento en enlaces. Cada JOIN es una ruta segura entre ruinas digitales. Aprende la disciplina: izquierda, derecha, completo, cruzado.

****Objetivo****

Usa RIGHT JOIN entre las tablas Bases y Survivors, pero con WHERE b.BaseID IS NULL para encontrar supervivientes sin base asignada. Selecciona solo s.Nombre. Deben aparecer 2 supervivientes huérfanos.

****Notas de inteligencia****

Esta consulta encuentra supervivientes sin base asignada (huérfanos): Maya Chen y Diego Morales.

****Consulta sugerida****

[SQL]

```
SELECT s.Nombre AS Superviviente FROM Bases b RIGHT JOIN Survivors s ON b.BaseID = s.BaseID WHERE b.BaseID IS NULL;
```

MISIÓN 18: ■ CROSS JOIN - PRODUCTO CARTESIANO

****Fecha:**** 29 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Entrenamiento en enlaces. Cada JOIN es una ruta segura entre ruinas digitales. Aprende la disciplina: izquierda, derecha, completo, cruzado.

****Objetivo****

Usa CROSS JOIN entre tablas Bases (b) y Survivors (s) sin condición ON. Selecciona b.Nombre como Base y s.Nombre como Superviviente. Creará TODAS las combinaciones posibles:
8 bases × 8 supervivientes = 64 filas.

****Notas de inteligencia****

CROSS JOIN crea todas las combinaciones posibles. 8 bases × 8 supervivientes = 64 combinaciones!

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, s.Nombre AS Superviviente FROM Bases b CROSS JOIN Survivors s;
```

MISIÓN 19: ■ MÚLTIPLES JOINS - TRES TABLAS

****Fecha:**** 30 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Entrenamiento en enlaces. Cada JOIN es una ruta segura entre ruinas digitales. Aprende la disciplina: izquierda, derecha, completo, cruzado.

****Objetivo****

Conecta tres tablas en una consulta: Survivors (s) INNER JOIN Bases (b) ON s.BaseID=b.BaseID, luego LEFT JOIN Resources (r) ON b.BaseID=r.BaseID. Selecciona s.Nombre, b.Nombre y r.ComidaRaciones. Usa LEFT JOIN para Resources porque no todas las

bases tienen recursos.

****Notas de inteligencia****

Usa múltiples JOINS para conectar tres tablas. LEFT JOIN para Resources porque no todas las bases tienen recursos.

****Consulta sugerida****

[SQL]

SELECT s.Nombre AS Superviviente, b.Nombre AS Base, r.ComidaRaciones FROM Survivors s INNER JOIN Bases b ON s.BaseID = b.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID;

MISIÓN 20: ■ AUTO-JOIN – LA TABLA CONSIGO MISMA

****Fecha:**** 31 de agosto de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Entrenamiento en enlaces. Cada JOIN es una ruta segura entre ruinas digitales. Aprende la disciplina: izquierda, derecha, completo, cruzado.

****Objetivo****

Usa alias s1 y s2 para hacer JOIN de la tabla Survivors consigo misma: s1 INNER JOIN s2 ON

s1.Edad = s2.Edad AND s1.SurvivorID < s2.SurvivorID. Selecciona s1.Nombre, s2.Nombre y s1.Edad. Esto encuentra parejas de supervivientes con la misma edad (evitando duplicados).

****Notas de inteligencia****

Auto-JOIN compara la tabla consigo misma. Deberías encontrar 2 pares: Marcus+Zara (41) y Li+Diego (27)

****Consulta sugerida****

[SQL]

SELECT s1.Nombre AS Superviviente1, s2.Nombre AS Superviviente2, s1.Edad FROM Survivors s1 INNER JOIN Survivors s2 ON s1.Edad = s2.Edad AND s1.SurvivorID < s2.SurvivorID;

MISIÓN 21: ■ EQUIPO POR BASE – ANÁLISIS REAL

****Fecha:**** 1 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que sea en nuestro tablero.

****Objetivo****

ATAPUERCA NET – CONFIDENCIAL

Usa JOIN entre tablas Survivors (s) y Bases (b) conectadas por BaseID. Selecciona s.Nombre, s.Rol, b.Nombre y b.TipoBase, ordenado por nombre de base. Deben aparecer 6 supervivientes con base asignada.

****Notas de inteligencia****

Verás que Fortaleza Norte tiene 3 supervivientes (Elena, Marcus, Hugo), y hay 1 superviviente en Cúpula Esperanza, Puesto Avanzado Alpha y Refugio Delta cada una

****Consulta sugerida****

[SQL]

```
SELECT s.Nombre AS Superviviente, s.Rol, b.Nombre AS Base, b.TipoBase FROM Survivors s JOIN Bases b ON s.BaseID = b.BaseID ORDER BY b.Nombre;
```

MISIÓN 22: ■ INVENTARIO COMPLETO POR BASE

****Fecha:**** 2 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
* Comando Central – General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que sea en nuestro tablero.

****Objetivo****

Usa LEFT JOIN entre tablas Bases (b) y Resources (r) conectadas por BaseID. Selecciona b.Nombre, y usa COALESCE para mostrar 0 en lugar de NULL para ComidaRaciones, AguaLitros y Armas. Ordena por nombre de base.

****Notas de inteligencia****

LEFT JOIN muestra todas las bases. COALESCE convierte NULL a 0 para bases sin recursos

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, COALESCE(r.ComidaRaciones, 0) AS Comida, COALESCE(r.AguaLitros, 0) AS Agua, COALESCE(r.Armas, 0) AS Armas FROM Bases b LEFT JOIN Resources r ON b.BaseID = r.BaseID ORDER BY b.Nombre;
```

MISIÓN 23: ■■ COMANDO Y SUBORDINADOS

****Fecha:**** 3 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
* Comando Central – General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que sea en nuestro tablero.

****Objetivo****

Usa JOIN entre tablas Bases (b) y Survivors (s) conectadas por BaseID, con WHERE b.EsComandoCentral = '1'. Selecciona b.Nombre, s.Nombre y s.Rol. Deben aparecer los 3 supervivientes del comando central.

****Notas de inteligencia****

La Fortaleza Norte es el comando central con Elena (Médica), Marcus (Soldado) y Hugo (Comandante)

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, s.Nombre AS Superviviente, s.Rol FROM Bases b JOIN Survivors s ON b.BaseID = s.BaseID WHERE b.EsComandoCentral = '1';
```

MISIÓN 24: ■ ESTADÍSTICAS POR HEMISFERIO

****Fecha:**** 4 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
* Comando Central – General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que sea en nuestro tablero.

****Objetivo****

Usa LEFT JOIN entre Bases (b) y Survivors (s), con CASE WHEN b.Latitud > 0 para clasificar hemisferios. Agrupa con GROUP BY y cuenta bases y supervivientes por hemisferio. Deben aparecer 2 filas: Norte y Sur.

****Notas de inteligencia****

Usa CASE WHEN para clasificar por hemisferio y GROUP BY para agrupar estadísticas

****Consulta sugerida****

[SQL]

```
SELECT CASE WHEN b.Latitud > 0 THEN 'Norte' ELSE 'Sur' END AS Hemisferio, COUNT(DISTINCT b.BaseID) AS NumBases, COUNT(s.SurvivorID) AS NumSupervivientes FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID GROUP BY CASE WHEN b.Latitud > 0 THEN 'Norte' ELSE 'Sur' END;
```

MISIÓN 25: ■■ ARSENAL POR SUPERVIVIENTE

****Fecha:**** 5 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
* Comando Central – General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que

sea en nuestro tablero.

****Objetivo****

Usa JOIN entre Survivors (s) y Bases (b), luego LEFT JOIN con Resources (r). Selecciona s.Nombre, s.Rol, b.Nombre y r.Armas (usa COALESCE para mostrar 0 si NULL). Ordena por armas descendente.

****Notas de inteligencia****

Combina 3 tablas para mostrar el armamento disponible para cada superviviente

****Consulta sugerida****

[SQL]

```
SELECT s.Nombre AS Superviviente, s.Rol, b.Nombre AS Base, COALESCE(r.Armas, 0) AS ArmasDisponibles FROM Survivors s JOIN Bases b ON s.BaseID = b.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID ORDER BY r.Armas DESC;
```

MISIÓN 26: ■ PROMEDIO DE EDAD POR BASE

****Fecha:**** 6 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
* Comando Central — General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que sea en nuestro tablero.

****Objetivo****

Usa JOIN entre Bases (b) y Survivors (s) conectadas por BaseID. Agrupa por b.Nombre y calcula COUNT de supervivientes y AVG de edad (usa CAST para convertir a FLOAT). Ordena por edad promedio descendente.

****Notas de inteligencia****

Fortaleza Norte debería tener la edad promedio más alta (Elena 34, Marcus 41, Hugo 39)

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, COUNT(s.SurvivorID) AS NumSupervivientes, AVG(CAST(s.Edad AS FLOAT)) AS EdadPromedio FROM Bases b JOIN Survivors s ON b.BaseID = s.BaseID GROUP BY b.Nombre ORDER BY EdadPromedio DESC;
```

MISIÓN 27: ■ LA BASE MÁS PRÓSPERA

****Fecha:**** 7 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
* Comando Central — General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que

sea en nuestro tablero.

****Objetivo****

Usa JOIN entre tablas Bases (b) y Resources (r) conectadas por BaseID. Suma las columnas ComidaRaciones + AguaLitros + Armas + Medicinas. Ordena por total descendente y usa LIMIT 1 para obtener solo la base con más recursos.

****Notas de inteligencia****

Suma todos los tipos de recursos para encontrar la base más próspera

****Consulta sugerida****

[SQL]

SELECT b.Nombre AS Base, r.ComidaRaciones + r.AguaLitros + r.Armas + r.Medicinas AS RecursosTotales FROM Bases b JOIN Resources r ON b.BaseID = r.BaseID ORDER BY RecursosTotales DESC LIMIT 1;

MISIÓN 28: ■ BASES SIN SUPERVIVIENTES

****Fecha:**** 8 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
* Comando Central – General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que sea en nuestro tablero.

****Objetivo****

Usa LEFT JOIN entre tablas Bases (b) y Survivors (s) conectadas por BaseID, con WHERE s.BaseID IS NULL para encontrar bases vacías. Selecciona b.Nombre y b.TipoBase. Deben aparecer 4 bases sin supervivientes.

****Notas de inteligencia****

LEFT JOIN con WHERE NULL encuentra bases vacías. Las bases IA (Torre Omega, Nido Central) no deberían tener supervivientes

****Consulta sugerida****

[SQL]

SELECT b.Nombre AS Base, b.TipoBase FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID WHERE s.BaseID IS NULL;

MISIÓN 29: ■ LIDERAZGO Y EXPERIENCIA

****Fecha:**** 9 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
* Comando Central – General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que

sea en nuestro tablero.

****Objetivo****

Usa JOIN entre Survivors (s) y Bases (b) conectadas por BaseID, con WHERE s.Rol IN ('Comandante', 'Médica'). Selecciona s.Nombre, s.Rol, s.Edad y b.Nombre. Ordena por edad descendente. Deben aparecer 3 líderes.

****Notas de inteligencia****

Filtra por roles de liderazgo. Zara es Comandante (41), Hugo es Comandante (39) y Elena es Médica (34)

****Consulta sugerida****

[SQL]

SELECT s.Nombre AS Superviviente, s.Rol, s.Edad, b.Nombre AS Base FROM Survivors s JOIN Bases b ON s.BaseID = b.BaseID WHERE s.Rol IN ('Comandante', 'Médica') ORDER BY s.Edad DESC;

MISIÓN 30: ■ BALANCE DE RECURSOS

****Fecha:**** 10 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que sea en nuestro tablero.

****Objetivo****

Usa JOIN entre Bases (b) y Resources (r) conectadas por BaseID. Selecciona b.Nombre, r.ComidaRaciones, r.AguaLitros, calcula la diferencia absoluta con ABS(), y usa CASE WHEN para clasificar el balance. Ordena por diferencia ascendente.

****Notas de inteligencia****

Usa ABS para diferencia absoluta y CASE WHEN para clasificar el tipo de balance

****Consulta sugerida****

[SQL]

SELECT b.Nombre AS Base, r.ComidaRaciones, r.AguaLitros, ABS(r.ComidaRaciones - r.AguaLitros) AS DiferenciaRecursos, CASE WHEN r.ComidaRaciones > r.AguaLitros THEN 'Más comida' WHEN r.AguaLitros > r.ComidaRaciones THEN 'Más agua' ELSE 'Equilibrado' END AS TipoBalance FROM Bases b JOIN Resources r ON b.BaseID = r.BaseID ORDER BY DiferenciaRecursos;

MISIÓN 31: ■ ESTADÍSTICAS GENERALES DEL ASENTAMIENTO

****Fecha:**** 11 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

****Objetivo****

Usa LEFT JOIN desde Bases (b) hacia Survivors (s) y Resources (r). Cuenta bases y supervivientes con COUNT(DISTINCT), suma recursos con SUM y COALESCE. Sin GROUP BY para obtener un resumen general en 1 fila.

****Notas de inteligencia****

Usa agregaciones para crear un dashboard completo de estadísticas

****Consulta sugerida****

[SQL]

```
SELECT COUNT(DISTINCT b.BaseID) AS TotalBases, COUNT(DISTINCT s.SurvivorID) AS TotalSupervivientes, SUM(COALESCE(r.ComidaRaciones, 0)) AS ComidaTotal, SUM(COALESCE(r.AguaLitros, 0)) AS AguaTotal, SUM(COALESCE(r.Armas, 0)) AS ArmasTotal FROM Bases b LEFT JOIN Survivor s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID;
```

MISIÓN 32: ■ RANKING DE BASES POR POBLACIÓN

****Fecha:**** 12 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

****Objetivo****

Usa LEFT JOIN entre Bases (b) y Survivors (s), agrupa por b.BaseID, b.Nombre, b.TipoBase. Cuenta supervivientes con COUNT(s.SurvivorID) y usa RANK() OVER para crear ranking. Ordena por población descendente.

****Notas de inteligencia****

Usa window functions con RANK() para crear un ranking

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, b.TipoBase, COUNT(s.SurvivorID) AS NumSupervivientes, RANK() OVER (ORDER BY COUNT(s.SurvivorID) DESC) AS Ranking FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID GROUP BY b.BaseID, b.Nombre, b.TipoBase ORDER BY NumSupervivientes DESC;
```

MISIÓN 33: ■ ANÁLISIS DE ROLES CRÍTICOS

****Fecha:**** 13 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
****** Comando Central — General Luna

****Contexto narrativo****

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

****Objetivo****

De la tabla Survivors, agrupa por s.Rol y calcula COUNT(*), AVG de edad (convertida a FLOAT), MIN edad y MAX edad. Ordena por cantidad descendente para identificar roles escasos.

****Notas de inteligencia****

Identifica qué roles son escasos. Solo hay 1 de cada rol excepto... ¿hay algún rol duplicado?

****Consulta sugerida****

[SQL]

```
SELECT s.Rol, COUNT(*) AS Cantidad, AVG(CAST(s.Edad AS FLOAT)) AS EdadPromedio, MIN(s.Edad) AS MasJoven, MAX(s.Edad) AS MasViejo FROM Survivors s GROUP BY s.Rol ORDER BY Cantidad DESC;
```

MISIÓN 34: ■ DENSIDAD POBLACIONAL POR BASE

****Fecha:**** 14 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
****** Comando Central — General Luna

****Contexto narrativo****

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

****Objetivo****

Usa LEFT JOIN entre Bases (b) y Survivors (s), filtra WHERE b.Capacidad IS NOT NULL. Agrupa por base y calcula porcentaje de ocupación usando COUNT / Capacidad * 100. Usa NULLIF para evitar división por cero.

****Notas de inteligencia****

NULLIF evita división por cero. Solo las bases humanas tienen capacidad definida

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, b.Capacidad, COUNT(s.SurvivorID) AS Ocupantes, ROUND((COUNT(s.SurvivorID) * 100.0 / NULLIF(b.Capacidad, 0)), 2) AS PorcentajeOcupacion FROM Bases b LEFT
```

ATAPUERCA NET – CONFIDENCIAL

JOIN Survivors s ON b.BaseID = s.BaseID WHERE b.Capacidad IS NOT NULL GROUP BY b.BaseID, b.Nombre, b.Capacidad ORDER BY PorcentajeOcupacion DESC;

MISIÓN 35: ■ ANÁLISIS DE AUTOSUFICIENCIA

Fecha: 15 de septiembre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:**
** Comando Central – General Luna

Contexto narrativo

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

Objetivo

Usa LEFT JOIN desde Bases (b) hacia Survivors (s) y Resources (r). Agrupa por base y calcula recursos per cápita usando CASE WHEN para evitar división por cero. Incluye COUNT supervivientes, COALESCE para recursos NULL.

Notas de inteligencia

Calcula recursos per cápita para identificar qué bases están mejor preparadas

Consulta sugerida

[SQL]

SELECT b.Nombre AS Base, COUNT(s.SurvivorID) AS Supervivientes, COALESCE(r.ComidaRaciones, 0) AS Comida, COALESCE(r.AguaLitros, 0) AS Agua, CASE WHEN COUNT(s.SurvivorID) = 0 THEN 0 ELSE ROUND(COALESCE(r.ComidaRaciones, 0) / CAST(COUNT(s.SurvivorID) AS FLOAT), 2) END AS ComidaPerCapita, CASE WHEN COUNT(s.SurvivorID) = 0 THEN 0 ELSE ROUND(COALESCE(r.AguaLitros, 0) / CAST(COUNT(s.SurvivorID) AS FLOAT), 2) END AS AguaPerCapita FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID GROUP BY b.BaseID, b.Nombre, r.ComidaRaciones, r.AguaLitros ORDER BY ComidaPerCapita DESC;

MISIÓN 36: ■ ANÁLISIS GEOESPACIAL

Fecha: 16 de septiembre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:**
** Comando Central – General Luna

Contexto narrativo

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

Objetivo

Usa CROSS JOIN en tabla Bases (b1, b2) con WHERE b1.BaseID < b2.BaseID para evitar duplicados. Selecciona nombres, coordenadas y calcula distancia aproximada con ABS de diferencias lat/lon. Ordena descendente, LIMIT 3.

****Notas de inteligencia****

CROSS JOIN compara cada base con todas las demás. Usa ABS para distancia Manhattan aproximada

****Consulta sugerida****

[SQL]

```
SELECT b1.Nombre AS Base1, b2.Nombre AS Base2, b1.Latitud AS Lat1, b1.Longitud AS Lon1, b2.Latitud AS Lat2, b2.Longitud AS Lon2, ABS(b1.Latitud - b2.Latitud) + ABS(b1.Longitud - b2.Longitud) AS DistanciaAproximada FROM Bases b1 CROSS JOIN Bases b2 WHERE b1.BaseID < b2.BaseID ORDER BY DistanciaAproximada DESC LIMIT 3;
```

MISIÓN 37: ■ ANÁLISIS DE EFICIENCIA OPERATIVA

****Fecha:**** 17 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

****Objetivo****

Usa LEFT JOIN desde Bases (b) hacia Survivors (s) y Resources (r). Agrupa por base y calcula un índice de eficiencia combinando: supervivientes*10 + recursos totales/10 + capacidad/5. Usa COALESCE para manejar NULL.

****Notas de inteligencia****

Combina múltiples factores con pesos diferentes para crear un índice compuesto de eficiencia

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, COUNT(s.SurvivorID) AS Supervivientes, COALESCE(r.ComidaRaciones + r.AguaLitros + r.Armas + r.Medicinas, 0) AS RecursosTotales, COALESCE(b.Capacidad, 0) AS Capacidad, ROUND((COUNT(s.SurvivorID) * 10 + COALESCE(r.ComidaRaciones + r.AguaLitros + r.Armas + r.Medicinas, 0) / 10.0 + COALESCE(b.Capacidad, 0) / 5.0), 2) AS IndiceEficiencia FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID GROUP BY b.BaseID, b.Nombre, r.ComidaRaciones, r.AguaLitros, r.Armas, r.Medicinas, b.Capacidad ORDER BY IndiceEficiencia DESC;
```

MISIÓN 38: ■ ANÁLISIS GENERACIONAL

****Fecha:**** 18 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

****Objetivo****

De la tabla Survivors, usa CASE WHEN para clasificar por rangos de edad (< 20, 20-29, 30-39, 40+). Agrupa por rango y calcula COUNT, AVG de edad, y STRING_AGG para concatenar nombres con roles. Ordena por edad promedio.

****Notas de inteligencia****

Usa CASE WHEN para clasificar por rangos de edad y STRING_AGG para concatenar nombres

****Consulta sugerida****

[SQL]

```
SELECT CASE WHEN s.Edad < 20 THEN 'Joven (< 20)' WHEN s.Edad < 30 THEN 'Adulto Joven (20-29)' WHEN s.Edad < 40 THEN 'Adulto (30-39)' ELSE 'Veterano (40+)' END AS GrupoEdad, COUNT(*) AS Cantidad, AVG(CAST(s.Edad AS FLOAT)) AS EdadPromedio, STRING_AGG(s.Nombre + ' (' + s.Rol + ')', ', ') AS Miembros FROM Survivors s GROUP BY CASE WHEN s.Edad < 20 THEN 'Joven (< 20)' WHEN s.Edad < 30 THEN 'Adulto Joven (20-29)' WHEN s.Edad < 40 THEN 'Adulto (30-39)' ELSE 'Veterano (40+)' END ORDER BY EdadPromedio;
```

MISIÓN 39: ■ MATRIZ DE INTERCAMBIO DE RECURSOS

****Fecha:**** 19 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

****Objetivo****

Usa JOIN entre Bases (b1) y Resources (r1), luego CROSS JOIN con Bases (b2) y Resources (r2), con WHERE b1.BaseID != b2.BaseID. Compara recursos de comida, calcula diferencias con ABS y clasifica tipo de intercambio con CASE WHEN.

****Notas de inteligencia****

CROSS JOIN crea todas las combinaciones para analizar posibles intercambios entre bases

****Consulta sugerida****

[SQL]

```
SELECT b1.Nombre AS BaseOrigen, b2.Nombre AS BaseDestino, r1.ComidaRaciones AS ComidaOrigen, r2.ComidaRaciones AS ComidaDestino, ABS(r1.ComidaRaciones - r2.ComidaRaciones) AS DiferenciaComida, CASE WHEN r1.ComidaRaciones > r2.ComidaRaciones THEN 'Puede enviar comida' WHEN r1.ComidaRaciones < r2.ComidaRaciones THEN 'Necesita comida' ELSE 'Equilibrado' END AS TipoIntercambio FROM Bases b1 JOIN Resources r1 ON b1.BaseID = r1.BaseID CROSS JOIN Bases b2 JOIN Resources r2 ON b2.BaseID = r2.BaseID WHERE b1.BaseID != b2.BaseID ORDER BY DiferenciaComida DESC;
```

MISIÓN 40: ■■ PLAN DE EXPANSIÓN ESTRATÉGICA

Fecha: 20 de septiembre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:**
 ** Comando Central – General Luna

Contexto narrativo

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

Objetivo

Usa CTE llamado BaseStats que hace LEFT JOIN desde Bases hacia Survivors y Resources, agrupa por TipoBase y calcula estadísticas promedio. Luego selecciona del CTE y agrega recomendaciones con CASE WHEN por tipo de base.

Notas de inteligencia

Usa CTE para calcular estadísticas por tipo de base y generar recomendaciones estratégicas

Consulta sugerida

[SQL]

```
WITH BaseStats AS (SELECT b.TipoBase, COUNT(*) AS NumBases, AVG(CAST(COUNT(s.SurvivorID)
AS FLOAT)) AS PromSupervivientes, AVG(CAST(COALESCE(r.ComidaRaciones + r.AguaLitros, 0) A
S FLOAT)) AS PromRecursos FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT
JOIN Resources r ON b.BaseID = r.BaseID GROUP BY b.TipoBase) SELECT TipoBase, NumBases, R
OUND(PromSupervivientes, 2) AS PromedioSupervivientes, ROUND(PromRecursos, 2) AS Promedio
Recursos, CASE WHEN TipoBase = 'Humana' THEN 'Expandir en zonas templadas' ELSE 'Monitore
ar actividad IA' END AS RecomendacionExpansion FROM BaseStats ORDER BY PromRecursos DESC;
```

MISIÓN 41: ■ MISIÓN CRÍTICA: BASE MÁS VULNERABLE

Fecha: 21 de septiembre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:**
 ** Comando Central – General Luna

Contexto narrativo

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

Objetivo

Encuentra la base con menos supervivientes pero que aún tenga gente asignada. Usa JOIN de Bases con Survivors, GROUP BY con HAVING que compare con subconsulta MIN. Debe mostrar nombre de base y número de supervivientes.

Notas de inteligencia

Usa subconsulta para encontrar el mínimo número de supervivientes y luego filtrar por esa

cantidad

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, COUNT(s.SurvivorID) AS NumSupervivientes FROM Bases b JOIN Survivors s ON b.BaseID = s.BaseID GROUP BY b.BaseID, b.Nombre HAVING COUNT(s.SurvivorID) = (SELECT MIN(cuenta) FROM (SELECT COUNT(s2.SurvivorID) AS cuenta FROM Bases b2 JOIN Survivors s2 ON b2.BaseID = s2.BaseID GROUP BY b2.BaseID) AS subconsulta);
```

MISIÓN 42: ■ ANÁLISIS COMPARATIVO DE RECURSOS

****Fecha:**** 22 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

****Objetivo****

De Bases (b) con JOIN a Resources (r), selecciona nombre y recursos totales. Filtra con WHERE donde la suma sea mayor que subconsulta AVG de la suma de todos los recursos. Ordena por recursos totales descendente.

****Notas de inteligencia****

Subconsulta en WHERE para comparar con el promedio. Las bases por encima del promedio son estratégicamente importantes

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, (r.ComidaRaciones + r.AguaLitros + r.Armas + r.Medicinas) AS RecursosTotales FROM Bases b JOIN Resources r ON b.BaseID = r.BaseID WHERE (r.ComidaRaciones + r.AguaLitros + r.Armas + r.Medicinas) > (SELECT AVG(ComidaRaciones + AguaLitros + Armas + Medicinas) FROM Resources) ORDER BY RecursosTotales DESC;
```

MISIÓN 43: ■ LIDERAZGO Y EXPERIENCIA CRÍTICA

****Fecha:**** 23 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

****Objetivo****

De Survivors (s) con JOIN a Bases (b), selecciona nombre, edad, rol y base. Filtra WHERE

edad > subconsulta AVG de todas las edades Y rol sea 'Comandante' o 'Médica'. Ordena por edad descendente.

****Notas de inteligencia****

Combina filtro de edad (subconsulta) con filtro de roles críticos. La experiencia es vital para la supervivencia

****Consulta sugerida****

[SQL]

```
SELECT s.Nombre, s.Edad, s.Rol, b.Nombre AS Base FROM Survivors s JOIN Bases b ON s.BaseID = b.BaseID WHERE s.Edad > (SELECT AVG(CAST(Edad AS FLOAT)) FROM Survivors) AND s.Rol IN ('Comandante', 'Médica') ORDER BY s.Edad DESC;
```

MISIÓN 44: ■ CAPACIDAD SUBUTILIZADA

****Fecha:**** 24 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

****Objetivo****

Usa LEFT JOIN de Bases (b) con Survivors (s), filtra WHERE capacidad IS NOT NULL. Agrupa por base y muestra nombre, capacidad, ocupantes actuales y espacio disponible (capacidad-ocupantes). HAVING para solo bases con espacio libre.

****Notas de inteligencia****

HAVING filtra grupos después de GROUP BY. Solo bases humanas tienen capacidad definida

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, b.Capacidad, COUNT(s.SurvivorID) AS Ocupantes, (b.Capacidad - COUNT(s.SurvivorID)) AS EspacioDisponible FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID WHERE b.Capacidad IS NOT NULL GROUP BY b.BaseID, b.Nombre, b.Capacidad HAVING b.Capacidad > COUNT(s.SurvivorID) ORDER BY EspacioDisponible DESC;
```

MISIÓN 45: ■ BASES AUTÓNOMAS VS DEPENDIENTES

****Fecha:**** 25 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

****Objetivo****

Usa LEFT JOIN de Bases (b) con Resources (r). Clasifica con CASE WHEN usando EXISTS para verificar si recursos >= 500 comida Y 500 agua ('Autónoma'), si existe cualquier recurso ('Dependiente'), sino 'Sin recursos'. Muestra también comida y agua con COALESCE.

****Notas de inteligencia****

EXISTS verifica si existe al menos un registro que cumpla la condición. Bases autónomas necesitan 500+ de comida y agua

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, CASE WHEN EXISTS (SELECT 1 FROM Resources r WHERE r.BaseID = b.BaseID AND r.ComidaRaciones >= 500 AND r.AguaLitros >= 500) THEN 'Autónoma' WHEN EXISTS (SELECT 1 FROM Resources r WHERE r.BaseID = b.BaseID) THEN 'Dependiente' ELSE 'Sin recursos' END AS NivelAutonomia, COALESCE(r.ComidaRaciones, 0) AS Comida, COALESCE(r.AguaLitros, 0) AS Agua FROM Bases b LEFT JOIN Resources r ON b.BaseID = r.BaseID ORDER BY CASE WHEN EXISTS (SELECT 1 FROM Resources r2 WHERE r2.BaseID = b.BaseID AND r2.ComidaRaciones >= 500 AND r2.AguaLitros >= 500) THEN 1 ELSE 2 END;
```

MISIÓN 46: ■ SUPERVIVIENTES ÚNICOS POR CARACTERÍSTICAS

****Fecha:**** 26 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
**** Comando Central – General Luna**

****Contexto narrativo****

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

****Objetivo****

De Survivors (s) con JOIN a Bases (b), selecciona nombre, edad, rol, base y característica especial. Usa CASE WHEN con subconsultas MAX/MIN para identificar edad extrema, o roles especiales ('Comandante', 'Médica'). Ordena por edad.

****Notas de inteligencia****

Múltiples subconsultas para identificar características únicas. Cada superviviente tiene un valor estratégico específico

****Consulta sugerida****

[SQL]

```
SELECT s.Nombre, s.Edad, s.Rol, b.Nombre AS Base, CASE WHEN s.Edad = (SELECT MAX(Edad) FROM Survivors) THEN 'Más veterano' WHEN s.Edad = (SELECT MIN(Edad) FROM Survivors) THEN 'Más joven' WHEN s.Rol = 'Comandante' THEN 'Líder militar' WHEN s.Rol = 'Médica' THEN 'SopORTE vital' ELSE 'Especialista' END AS Caracteristica FROM Survivors s JOIN Bases b ON s.BaseID = b.BaseID ORDER BY s.Edad;
```

MISIÓN 47: ■ ANÁLISIS DE CONCENTRACIÓN POBLACIONAL

Fecha: 27 de septiembre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:**
 ** Comando Central – General Luna

Contexto narrativo

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

Objetivo

Usa dos CTEs: BasePopulation (LEFT JOIN Bases-Survivors agrupado) y StatsGenerales (AVG de supervivientes). Luego CROSS JOIN entre ambos CTEs y clasifica con CASE WHEN si sobrepoblada (>1.5x promedio), subpoblada (<0.5x), abandonada (0) o equilibrada.

Notas de inteligencia

CTE (Common Table Expression) para cálculos complejos. Analiza la distribución poblacional para detectar desequilibrios

Consulta sugerida

[SQL]

```
WITH BasePopulation AS (SELECT b.BaseID, b.Nombre, COUNT(s.SurvivorID) AS NumSupervivientes
FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID GROUP BY b.BaseID, b.Nombre),
StatsGenerales AS (SELECT AVG(CAST(NumSupervivientes AS FLOAT)) AS PromedioSupervivientes
FROM BasePopulation WHERE NumSupervivientes > 0) SELECT bp.Nombre AS Base, bp.NumSupervivientes,
sg.PromedioSupervivientes, CASE WHEN bp.NumSupervivientes > sg.PromedioSupervivientes * 1.5 THEN 'Sobrepoblada'
WHEN bp.NumSupervivientes < sg.PromedioSupervivientes * 0.5 AND bp.NumSupervivientes > 0 THEN 'Subpoblada'
WHEN bp.NumSupervivientes = 0 THEN 'Abandonada' ELSE 'Equilibrada' END AS EstadoPoblacional
FROM BasePopulation bp CROSS JOIN StatsGenerales sg ORDER BY bp.NumSupervivientes DESC;
```

MISIÓN 48: ■■ EVALUACIÓN DE SEGURIDAD POR ZONA

Fecha: 28 de septiembre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:**
 ** Comando Central – General Luna

Contexto narrativo

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

Objetivo

Agrupar por zona geográfica usando CASE WHEN con latitud (>42 Norte, >35 Central, resto Sur). LEFT JOIN Bases-Survivors-Resources, calcular COUNT bases, supervivientes, SUM/AVG armas. Clasificar seguridad según ratio armas/supervivientes.

Notas de inteligencia

Agrupar por zonas geográficas y calcular ratios de seguridad. Una persona por arma es el ideal mínimo

****Consulta sugerida****

[SQL]

```
SELECT CASE WHEN b.Latitud > 42 THEN 'Zona Norte' WHEN b.Latitud > 35 THEN 'Zona Central'
ELSE 'Zona Sur' END AS AreaGeografica, COUNT(DISTINCT b.BaseID) AS NumBases, COUNT(s.SurvivorID) AS TotalSupervivientes, SUM(COALESCE(r.Armas, 0)) AS ArmasTotal, ROUND(AVG(CAST(COALESCE(r.Armas, 0) AS FLOAT)), 2) AS ArmasPromedioPorBase, CASE WHEN SUM(COALESCE(r.Armas, 0)) >= COUNT(s.SurvivorID) THEN 'Bien armada' WHEN SUM(COALESCE(r.Armas, 0)) >= COUNT(s.SurvivorID) * 0.5 THEN 'Armamento moderado' ELSE 'Vulnerable' END AS NivelSeguridad FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID GROUP BY CASE WHEN b.Latitud > 42 THEN 'Zona Norte' WHEN b.Latitud > 35 THEN 'Zona Central' ELSE 'Zona Sur' END ORDER BY ArmasTotal DESC;
```

MISIÓN 49: ■ SIMULACIÓN DE EVACUACIÓN DE EMERGENCIA

****Fecha:**** 29 de septiembre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

****Objetivo****

Dos CTEs: BasesOrigen (JOIN Bases-Survivors agrupado), BasesDestino (LEFT JOIN con HAVING para espacio libre). CROSS JOIN entre CTEs con WHERE origen≠destino. Muestra plan evacuación según capacidad disponible vs personas a evacuar.

****Notas de inteligencia****

Dos CTEs para separar bases origen (con gente) y destino (con espacio). CROSS JOIN simula todas las posibilidades

****Consulta sugerida****

[SQL]

```
WITH BasesOrigen AS (SELECT b.BaseID, b.Nombre, COUNT(s.SurvivorID) AS Ocupantes FROM Bases b JOIN Survivors s ON b.BaseID = s.BaseID GROUP BY b.BaseID, b.Nombre), BasesDestino AS (SELECT b.BaseID, b.Nombre, b.Capacidad, COALESCE(COUNT(s.SurvivorID), 0) AS OcupantesActuales, (b.Capacidad - COALESCE(COUNT(s.SurvivorID), 0)) AS EspacioDisponible FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID WHERE b.Capacidad IS NOT NULL GROUP BY b.BaseID, b.Nombre, b.Capacidad HAVING b.Capacidad > COALESCE(COUNT(s.SurvivorID), 0)) SELECT bo.Nombre AS BaseOrigen, bo.Ocupantes AS PersonasAEvacuar, bd.Nombre AS BaseDestino, bd.EspacioDisponible, CASE WHEN bd.EspacioDisponible >= bo.Ocupantes THEN 'Evacuación completa posible' ELSE 'Evacuación parcial (' + CAST(bd.EspacioDisponible AS VARCHAR) + ' personas)' END AS PlanEvacuacion FROM BasesOrigen bo CROSS JOIN BasesDestino bd WHERE bo.BaseID != bd.BaseID ORDER BY bo.Ocupantes DESC, bd.EspacioDisponible DESC;
```

MISIÓN 50: ■ EVALUACIÓN INTEGRAL DE SUPERVIVENCIA

Fecha: 30 de septiembre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:**
 ** Comando Central – General Luna

Contexto narrativo

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

Objetivo

Crea índice de supervivencia combinando múltiples factores con LEFT JOIN de Bases hacia Survivors y Resources. Calcula puntuación ponderada: supervivientesxpeso + recursosxpeso + capacidadxpeso + factores geográficos.

Notas de inteligencia

CTE mega-complejo que integra población, recursos, liderazgo y tipo de base en un índice único de supervivencia

Consulta sugerida

[SQL]

```
WITH AnalisisCompleto AS (SELECT b.BaseID, b.Nombre AS Base, b.TipoBase, COUNT(s.SurvivorID) AS NumSupervivientes, COALESCE(b.Capacidad, 0) AS Capacidad, COALESCE(r.ComidaRaciones + r.AguaLitros + r.Armas + r.Medicinas, 0) AS RecursosTotales, CASE WHEN COUNT(s.SurvivorID) = 0 THEN 0 ELSE COALESCE(r.ComidaRaciones + r.AguaLitros + r.Armas + r.Medicinas, 0) / CAST(COUNT(s.SurvivorID) AS FLOAT) END AS RecursosPerCapita, COUNT(CASE WHEN s.Rol IN ('Comandante', 'Médica') THEN 1 END) AS LiderazgoPresente FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID GROUP BY b.BaseID, b.Nombre, b.TipoBase, b.Capacidad, r.ComidaRaciones, r.AguaLitros, r.Armas, r.Medicinas) SELECT Base, TipoBase, NumSupervivientes, RecursosTotales, ROUND(RecursosPerCapita, 2) AS RecursosPerCapita, LiderazgoPresente, ROUND((NumSupervivientes * 20 + RecursosTotales / 10.0 + RecursosPerCapita * 5 + LiderazgoPresente * 50 + CASE WHEN TipoBase = 'Humana' THEN 25 ELSE 0 END), 2) AS IndiceSupervivencia, CASE WHEN (NumSupervivientes * 20 + RecursosTotales / 10.0 + RecursosPerCapita * 5 + LiderazgoPresente * 50 + CASE WHEN TipoBase = 'Humana' THEN 25 ELSE 0 END) >= 200 THEN '■ Excelente' WHEN (NumSupervivientes * 20 + RecursosTotales / 10.0 + RecursosPerCapita * 5 + LiderazgoPresente * 50 + CASE WHEN TipoBase = 'Humana' THEN 25 ELSE 0 END) >= 100 THEN '■ Aceptable' ELSE '■ Crítico' END AS EstadoSupervivencia FROM AnalisisCompleto ORDER BY IndiceSupervivencia DESC;
```

MISIÓN 51: ■ COMANDO CENTRAL: ANÁLISIS DE PODER

Fecha: 1 de octubre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:** C
 omando Central – General Luna

Contexto narrativo

Operación maestra. CTEs encadenadas y simulaciones para decidir el futuro del asentamiento.

****Objetivo****

Dos CTEs: ComandoCentral (LEFT JOIN múltiple, filtra WHERE EsComandoCentral='1', agrupa y calcula liderazgo/recursos) y StatsGlobales (promedio personal). CROSS JOIN entre CTEs, compara comando vs estadísticas globales.

****Notas de inteligencia****

CTE múltiple para analizar el comando central vs estadísticas globales. El liderazgo y personal son críticos

****Consulta sugerida****

[SQL]

```
WITH ComandoCentral AS (SELECT b.BaseID, b.Nombre, COUNT(s.SurvivorID) AS PersonalTotal,
SUM(CASE WHEN s.Rol IN ('Comandante', 'Médica') THEN 1 ELSE 0 END) AS LiderazgoTotal, AVG
(CAST(s.Edad AS FLOAT)) AS EdadPromedio, COALESCE(r.ComidaRaciones + r.AguaLitros + r.Armas
+ r.Medicinas, 0) AS RecursosTotales FROM Bases b LEFT JOIN Survivors s ON b.BaseID =
s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID WHERE b.EsComandoCentral = '1' GROU
P BY b.BaseID, b.Nombre, r.ComidaRaciones, r.AguaLitros, r.Armas, r.Medicinas), StatsGlob
ales AS (SELECT AVG(CAST(NumSupervivientes AS FLOAT)) AS PromedioPersonalGlobal FROM (SEL
ECT COUNT(s.SurvivorID) AS NumSupervivientes FROM Bases b LEFT JOIN Survivors s ON b.Base
ID = s.BaseID GROUP BY b.BaseID) AS subconsulta WHERE NumSupervivientes > 0) SELECT cc.No
mbre AS ComandoCentral, cc.PersonalTotal, cc.LiderazgoTotal, ROUND(cc.EdadPromedio, 1) AS
EdadPromedio, cc.RecursosTotales, sg.PromedioPersonalGlobal, ROUND((cc.PersonalTotal / s
g.PromedioPersonalGlobal) * 100, 1) AS PorcentajeSuperioridad, CASE WHEN cc.LiderazgoTota
l >= 2 AND cc.PersonalTotal >= sg.PromedioPersonalGlobal * 1.5 THEN '■ Comando sólido' WH
EN cc.LiderazgoTotal >= 1 AND cc.PersonalTotal >= sg.PromedioPersonalGlobal THEN '■ Coman
do funcional' ELSE '■ Comando débil' END AS EvaluacionComando FROM ComandoCentral cc CROS
S JOIN StatsGlobales sg;
```

MISIÓN 52: ■ PREDICCIÓN DE ESCASEZ DE RECURSOS

****Fecha:**** 2 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** C
omando Central – General Luna

****Contexto narrativo****

Operación maestra. CTEs encadenadas y simulaciones para decidir el futuro del asentamiento.

****Objetivo****

CTE ConsumoSimulado: LEFT JOIN Bases-Survivors-Resources, calcula consumo diario (3 comida + 5 agua por superviviente). Proyecta duración recursos con división, usa NULLIF para evitar error, clasifica alertas según días restantes.

****Notas de inteligencia****

Simula consumo diario (3 raciones comida, 5L agua por persona) para predecir escasez. NULLIF evita división por cero

****Consulta sugerida****

[SQL]

```
WITH ConsumoSimulado AS (SELECT b.Nombre AS Base, COUNT(s.SurvivorID) AS Consumidores, COALESCE(r.ComidaRaciones, 0) AS ComidaDisponible, COALESCE(r.AguaLitros, 0) AS AguaDisponible, COUNT(s.SurvivorID) * 3 AS ConsumoComidaDiario, COUNT(s.SurvivorID) * 5 AS ConsumoAguaDiario FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID WHERE COUNT(s.SurvivorID) > 0 GROUP BY b.BaseID, b.Nombre, r.ComidaRaciones, r.AguaLitros) SELECT Base, Consumidores, ComidaDisponible, AguaDisponible, ConsumoComidaDiario, ConsumoAguaDiario, CASE WHEN ConsumoComidaDiario = 0 THEN 'N/A' ELSE CAST(ComidaDisponible / ConsumoComidaDiario AS VARCHAR) + ' días' END AS DuracionComida, CASE WHEN ConsumoAguaDiario = 0 THEN 'N/A' ELSE CAST(AguaDisponible / ConsumoAguaDiario AS VARCHAR) + ' días' END AS DuracionAgua, CASE WHEN ComidaDisponible / NULLIF(ConsumoComidaDiario, 0) < 30 OR AguaDisponible / NULLIF(ConsumoAguaDiario, 0) < 30 THEN '■ Crisis inminente (<30 días)' WHEN ComidaDisponible / NULLIF(ConsumoComidaDiario, 0) < 60 OR AguaDisponible / NULLIF(ConsumoAguaDiario, 0) < 60 THEN '■ Situación tensa (<60 días)' ELSE '■ Recursos suficientes' END AS AlertaEscasez FROM ConsumoSimulado ORDER BY LEAST(ComidaDisponible / NULLIF(ConsumoComidaDiario, 0), AguaDisponible / NULLIF(ConsumoAguaDiario, 0));
```

MISIÓN 53: ■■ SIMULACIÓN DE CONFLICTO: PREPARACIÓN MILITAR

****Fecha:**** 3 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación maestra. CTEs encadenadas y simulaciones para decidir el futuro del asentamiento.

****Objetivo****

Dos CTEs: CapacidadMilitar (GROUP BY base, cuenta personal militar y armas) y EvaluacionDefensa (calcula puntuación según algoritmo: militar×2+armas si bien armados). UNION ALL agrega fila resumen total del asentamiento.

****Notas de inteligencia****

Algoritmo militar complejo: soldados entrenados valen x2, armas sin soldados valen la mitad. UNION ALL agrega resumen total

****Consulta sugerida****

[SQL]

```
WITH CapacidadMilitar AS (SELECT b.Nombre AS Base, COUNT(s.SurvivorID) AS PersonalTotal, COUNT(CASE WHEN s.Rol IN ('Soldado', 'Comandante') THEN 1 END) AS PersonalMilitar, COALESCE(r.Armas, 0) AS ArmasDisponibles, AVG(CAST(s.Edad AS FLOAT)) AS EdadPromedio FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID GROUP BY b.BaseID, b.Nombre, r.Armas), EvaluacionDefensa AS (SELECT Base, PersonalTotal, PersonalMilitar, ArmasDisponibles, ROUND(EdadPromedio, 1) AS EdadPromedio, CASE WHEN PersonalMilitar > 0 AND ArmasDisponibles >= PersonalMilitar THEN PersonalMilitar * 2 + ArmasDisponibles WHEN PersonalMilitar > 0 THEN PersonalMilitar + ArmasDisponibles ELSE ArmasDisponibles / 2.0 END AS PuntuacionDefensa FROM CapacidadMilitar) SELECT Base, PersonalTotal, PersonalMilitar, ArmasDisponibles, EdadPromedio, ROUND(PuntuacionDefensa, 1) AS Cap
```

```

acidadDefensiva, CASE WHEN PuntuacionDefensa >= 20 THEN '■ Fortaleza defensiva' WHEN PuntuacionDefensa >= 10 THEN '■ Defensa moderada' WHEN PuntuacionDefensa >= 5 THEN '■ Defensa básica' ELSE '■ Indefensa' END AS NivelDefensa FROM EvaluacionDefensa UNION ALL SELECT 'TOTAL ASENTAMIENTO' AS Base, SUM(PersonalTotal) AS PersonalTotal, SUM(PersonalMilitar) AS PersonalMilitar, SUM(ArmasDisponibles) AS ArmasDisponibles, AVG(EdadPromedio) AS EdadPromedio, SUM(PuntuacionDefensa) AS CapacidadDefensiva, CASE WHEN SUM(PuntuacionDefensa) >= 50 THEN '■ Asentamiento fortificado' WHEN SUM(PuntuacionDefensa) >= 25 THEN '■ Defensa coordinable' ELSE '■ Vulnerable a ataques' END AS NivelDefensa FROM EvaluacionDefensa ORDER BY CapacidadDefensiva DESC;

```

MISIÓN 54: ■ ANÁLISIS DE DIVERSIDAD DE HABILIDADES

****Fecha:**** 4 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación maestra. CTes encadenadas y simulaciones para decidir el futuro del asentamiento.

****Objetivo****

Dos CTes: AnalisisRoles (GROUP BY rol, STRING_AGG para listas) y CriticidadRol (evalúa riesgo por cantidad: 1=crítico, 2=vulnerable, >2=estable). UNION ALL agrega fila resumen con COUNT DISTINCT roles y evaluación diversidad.

****Notas de inteligencia****

Análisis complejo de roles únicos, redundancia y riesgos. STRING_AGG concatena listas de personas por rol

****Consulta sugerida****

[SQL]

```

WITH AnalisisRoles AS (SELECT s.Rol, COUNT(*) AS Cantidad, AVG(CAST(s.Edad AS FLOAT)) AS EdadPromedio, MIN(s.Edad) AS EdadMinima, MAX(s.Edad) AS EdadMaxima, STRING_AGG(s.Nombre + ' (' + CAST(s.Edad AS VARCHAR) + ')', ', ') AS ListaPersonas FROM Survivors s GROUP BY s.Rol), CriticidadRol AS (SELECT Rol, Cantidad, ROUND(EdadPromedio, 1) AS EdadPromedio, EdadMinima, EdadMaxima, ListaPersonas, CASE WHEN Cantidad = 1 THEN '■ CRÍTICO - Punto único de fallo' WHEN Cantidad = 2 THEN '■ VULNERABLE - Redundancia mínima' ELSE '■ ESTABLE - Múltiple cobertura' END AS NivelRiesgo, CASE WHEN EdadPromedio > 35 THEN '■■ Envejecimiento' WHEN EdadPromedio < 25 THEN '■ Joven' ELSE '■ Maduro' END AS PerfilEdad FROM AnalisisRoles) SELECT Rol, Cantidad, EdadPromedio, EdadMinima, EdadMaxima, NivelRiesgo, PerfilEdad, ListaPersonas FROM CriticidadRol UNION ALL SELECT 'RESUMEN DIVERSIDAD' AS Rol, COUNT(DISTINCT cr.Rol) AS RolesUnicos, AVG(cr.EdadPromedio) AS EdadGlobalPromedio, MIN(cr.EdadMinima) AS EdadMinimaGlobal, MAX(cr.EdadMaxima) AS EdadMaximaGlobal, CASE WHEN AVG(cr.Cantidad) < 1.5 THEN '■ Especialización excesiva' WHEN COUNT(DISTINCT cr.Rol) >= 4 THEN '■ Diversidad saludable' ELSE '■ Diversidad limitada' END AS EvaluacionDiversidad, CASE WHEN AVG(cr.EdadPromedio) BETWEEN 25 AND 40 THEN '■ Rango óptimo' ELSE '■■ Desequilibrio generacional' END AS BalanceGeneracional, 'Roles únicos: ' + CAST(COUNT(DISTINCT cr.Rol) AS VARCHAR) + ' | Promedio personas/rol: ' + CAST(ROUND(AVG(cr.Cantidad), 1) AS VARCHAR) AS EstadisticasGenerales FROM CriticidadRol cr ORDER BY CASE WHEN Rol = 'RESUMEN DIVERSIDAD' THEN

```

1 ELSE 0 END, NivelRiesgo DESC;

MISIÓN 55: ■ OPTIMIZACIÓN GEOESPACIAL DE RUTAS

****Fecha:**** 5 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación maestra. CTEs encadenadas y simulaciones para decidir el futuro del asentamiento.

****Objetivo****

Tres CTEs: DistanciasEntreBases (CROSS JOIN con distancia euclidiana), CentroGeografico (AVG coordenadas), BasesCentrales (distancia al centro). UNION ALL múltiple para centro, base central, rutas extremas.

****Notas de inteligencia****

Geometría espacial con múltiples CTEs. Calcula centro geográfico, base central, y rutas extremas para optimización logística

****Consulta sugerida****

[SQL]

```
WITH DistanciasEntreBases AS (SELECT b1.Nombre AS BaseOrigen, b2.Nombre AS BaseDestino, b1.Latitud AS Lat1, b1.Longitud AS Lon1, b2.Latitud AS Lat2, b2.Longitud AS Lon2, ROUND(SQRT(POWER(b1.Latitud - b2.Latitud, 2) + POWER(b1.Longitud - b2.Longitud, 2)), 4) AS DistanciaEuclidiana FROM Bases b1 CROSS JOIN Bases b2 WHERE b1.BaseID != b2.BaseID), CentroGeografico AS (SELECT AVG(Latitud) AS LatitudCentral, AVG(Longitud) AS LongitudCentral FROM Bases), BasesCentrales AS (SELECT b.Nombre AS Base, b.Latitud, b.Longitud, cg.LatitudCentral, cg.LongitudCentral, ROUND(SQRT(POWER(b.Latitud - cg.LatitudCentral, 2) + POWER(b.Longitud - cg.LongitudCentral, 2)), 4) AS DistanciaAlCentro FROM Bases b CROSS JOIN CentroGeografico cg) SELECT 'CENTRO GEOGRÁFICO' AS TipoAnálisis, 'Lat: ' + CAST(ROUND(LatitudCentral, 4) AS VARCHAR) + ', Lon: ' + CAST(ROUND(LongitudCentral, 4) AS VARCHAR) AS Coordenadas, NULL AS Base, NULL AS DistanciaAlCentro, NULL AS EstrategiaLogistica FROM CentroGeografico UNION ALL SELECT 'BASE MÁS CENTRAL' AS TipoAnálisis, 'Punto logístico óptimo' AS Coordenadas, Base, DistanciaAlCentro, 'Hub de distribución ideal' AS EstrategiaLogistica FROM BasesCentrales WHERE DistanciaAlCentro = (SELECT MIN(DistanciaAlCentro) FROM BasesCentrales) UNION ALL SELECT 'RUTA MÁS LARGA' AS TipoAnálisis, BaseOrigen + ' → ' + BaseDestino AS Coordenadas, NULL AS Base, DistanciaEuclidiana AS DistanciaAlCentro, 'Ruta crítica a proteger' AS EstrategiaLogistica FROM DistanciasEntreBases WHERE DistanciaEuclidiana = (SELECT MAX(DistanciaEuclidiana) FROM DistanciasEntreBases) UNION ALL SELECT 'RUTA MÁS CORTA' AS TipoAnálisis, BaseOrigen + ' → ' + BaseDestino AS Coordenadas, NULL AS Base, DistanciaEuclidiana AS DistanciaAlCentro, 'Conexión rápida disponible' AS EstrategiaLogistica FROM DistanciasEntreBases WHERE DistanciaEuclidiana = (SELECT MIN(DistanciaEuclidiana) FROM DistanciasEntreBases) ORDER BY TipoAnálisis;
```

MISIÓN 56: ■ SISTEMA DE SALUD: CAPACIDAD MÉDICA

Fecha: 6 de octubre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:** C
omando Central – General Luna

****Contexto narrativo****

Operación maestra. CTEs encadenadas y simulaciones para decidir el futuro del asentamiento.

****Objetivo****

Dos CTEs: CapacidadMedica (COUNT médicos y medicinas) y EvaluacionSanitaria (ratios cobertura y per cápita). Calcula porcentaje cobertura médica y capacidad emergencia. UNIO
N
ALL agrega resumen sistema general.

****Notas de inteligencia****

Análisis sanitario complejo: ratio médico-población, medicinas per cápita, capacidad de emergencia. Incluye evaluación del sistema global

****Consulta sugerida****

[SQL]

```
WITH CapacidadMedica AS (SELECT b.Nombre AS Base, COUNT(s.SurvivorID) AS PoblacionTotal,
COUNT(CASE WHEN s.Rol = 'Médica' THEN 1 END) AS PersonalMedico, COALESCE(r.Medicinas, 0)
AS MedicinasDisponibles, COUNT(s.SurvivorID) AS PacientesPotenciales FROM Bases b LEFT JO
IN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID GROUP
BY b.BaseID, b.Nombre, r.Medicinas), EvaluacionSanitaria AS (SELECT Base, PoblacionTotal,
PersonalMedico, MedicinasDisponibles, PacientesPotenciales, CASE WHEN PersonalMedico = 0
THEN 0 WHEN PoblacionTotal = 0 THEN 0 ELSE ROUND((PersonalMedico * 100.0) / PoblacionTot
al, 2) END AS PorcentajeCoberturaMedica, CASE WHEN PoblacionTotal = 0 THEN 0 ELSE ROUND(M
edicinasDisponibles / CAST(PoblacionTotal AS FLOAT), 2) END AS MedicinasPerCapita, CASE W
HEN PersonalMedico = 0 AND PoblacionTotal > 0 THEN '■ SIN COBERTURA MÉDICA' WHEN Personal
Medico > 0 AND (PoblacionTotal / CAST(PersonalMedico AS FLOAT)) <= 5 THEN '■ COBERTURA EX
CELENTE' WHEN PersonalMedico > 0 AND (PoblacionTotal / CAST(PersonalMedico AS FLOAT)) <=
10 THEN '■ COBERTURA ADECUADA' WHEN PersonalMedico > 0 THEN '■ COBERTURA LIMITADA' ELSE '
■ BASE VACÍA' END AS NivelCobertura FROM CapacidadMedica) SELECT Base, PoblacionTotal, Pe
rsonalMedico, MedicinasDisponibles, PorcentajeCoberturaMedica, MedicinasPerCapita, NivelC
obertura, CASE WHEN MedicinasPerCapita >= 50 AND PersonalMedico > 0 THEN '■ Preparada par
a emergencias' WHEN MedicinasPerCapita >= 20 AND PersonalMedico > 0 THEN '■■ Capacidad bá
sica' WHEN PersonalMedico > 0 THEN '■ Recursos insuficientes' ELSE '■ Incapaz de responde
r' END AS CapacidadEmergencia FROM EvaluacionSanitaria UNION ALL SELECT 'SISTEMA GENERAL'
AS Base, SUM(PoblacionTotal) AS PoblacionTotal, SUM(PersonalMedico) AS PersonalMedico, S
UM(MedicinasDisponibles) AS MedicinasDisponibles, CASE WHEN SUM(PoblacionTotal) = 0 THEN
0 ELSE ROUND((SUM(PersonalMedico) * 100.0) / SUM(PoblacionTotal), 2) END AS CoberturaGene
ral, CASE WHEN SUM(PoblacionTotal) = 0 THEN 0 ELSE ROUND(SUM(MedicinasDisponibles) / CAST
(SUM(PoblacionTotal) AS FLOAT), 2) END AS MedicinasPerCapitaGeneral, CASE WHEN SUM(Person
alMedico) = 0 THEN '■ CRISIS SANITARIA TOTAL' WHEN (SUM(PoblacionTotal) / CAST(SUM(Person
alMedico) AS FLOAT)) <= 5 THEN '■ SISTEMA ROBUSTO' WHEN (SUM(PoblacionTotal) / CAST(SUM(P
ersonalMedico) AS FLOAT)) <= 10 THEN '■ SISTEMA FUNCIONAL' ELSE '■ SISTEMA SOBRECARGADO'
END AS EstadoSistema, CASE WHEN (SUM(MedicinasDisponibles) / CAST(SUM(PoblacionTotal) AS
FLOAT)) >= 50 AND SUM(PersonalMedico) > 0 THEN '■ Listo para pandemia' ELSE '■ Vulnerable
a crisis sanitaria' END AS PreparacionCrisis FROM EvaluacionSanitaria WHERE PoblacionTot
```

```
al > 0 ORDER BY CASE WHEN Base = 'SISTEMA GENERAL' THEN 1 ELSE 0 END, CapacidadEmergencia
DESC;
```

MISIÓN 57: ■ SIMULACIÓN DE INTERCAMBIO ECONÓMICO

****Fecha:**** 7 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación maestra. CTEs encadenadas y simulaciones para decidir el futuro del asentamiento.

****Objetivo****

Tres CTEs: PerfilesBase (recursos per cápita), IntercambiosOptimos (CROSS JOIN calcula transferencias cuando diferencia >50), RankingIntercambios (ROW_NUMBER ordena por beneficio). UNION ALL agrega resumen económico.

****Notas de inteligencia****

Economía simulada compleja: identifica desequilibrios de recursos, calcula transferencias óptimas y clasifica beneficios por niveles

****Consulta sugerida****

[SQL]

```
WITH PerfilesBase AS (SELECT b.BaseID, b.Nombre AS Base, COUNT(s.SurvivorID) AS Poblacion
, COALESCE(r.ComidaRaciones, 0) AS Comida, COALESCE(r.AguaLitros, 0) AS Agua, COALESCE(r.
Armas, 0) AS Armas, COALESCE(r.Medicinas, 0) AS Medicinas, CASE WHEN COUNT(s.SurvivorID)
= 0 THEN 0 ELSE COALESCE(r.ComidaRaciones, 0) / CAST(COUNT(s.SurvivorID) AS FLOAT) END AS
ComidaPerCapita, CASE WHEN COUNT(s.SurvivorID) = 0 THEN 0 ELSE COALESCE(r.AguaLitros, 0)
/ CAST(COUNT(s.SurvivorID) AS FLOAT) END AS AguaPerCapita FROM Bases b LEFT JOIN Survivo
rs s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID GROUP BY b.BaseI
D, b.Nombre, r.ComidaRaciones, r.AguaLitros, r.Armas, r.Medicinas), IntercambiosOptimos A
S (SELECT pl.Base AS BaseOrigen, p2.Base AS BaseDestino, pl.Poblacion AS PoblacionOrigen,
p2.Poblacion AS PoblacionDestino, CASE WHEN pl.ComidaPerCapita > p2.ComidaPerCapita + 50
THEN ROUND((pl.ComidaPerCapita - p2.ComidaPerCapita) / 2 * pl.Poblacion, 0) ELSE 0 END A
S ComidaATransferir, CASE WHEN pl.AguaPerCapita > p2.AguaPerCapita + 50 THEN ROUND((pl.Ag
uaPerCapita - p2.AguaPerCapita) / 2 * pl.Poblacion, 0) ELSE 0 END AS AguaATransferir, ABS
(pl.ComidaPerCapita - p2.ComidaPerCapita) + ABS(pl.AguaPerCapita - p2.AguaPerCapita) AS P
otencialBeneficio FROM PerfilesBase p1 CROSS JOIN PerfilesBase p2 WHERE p1.BaseID != p2.B
aseID AND p1.Poblacion > 0 AND p2.Poblacion > 0), RankingIntercambios AS (SELECT BaseOrig
en, BaseDestino, ComidaATransferir, AguaATransferir, PotencialBeneficio, ROW_NUMBER() OVE
R (PARTITION BY BaseOrigen ORDER BY PotencialBeneficio DESC) AS RankingBeneficio FROM Int
ercambiosOptimos WHERE ComidaATransferir > 0 OR AguaATransferir > 0) SELECT BaseOrigen, B
aseDestino, ComidaATransferir, AguaATransferir, ROUND(PotencialBeneficio, 2) AS Potencial
Beneficio, CASE WHEN ComidaATransferir > AguaATransferir THEN '■ Intercambio de comida pr
ioritario' WHEN AguaATransferir > ComidaATransferir THEN '■ Intercambio de agua prioritar
io' ELSE '■■■ Intercambio equilibrado' END AS TipoIntercambio, CASE WHEN PotencialBenefici
o > 100 THEN '■ Altamente beneficioso' WHEN PotencialBeneficio > 50 THEN '■ Moderadamente
beneficioso' ELSE '■ Beneficio mínimo' END AS NivelBeneficio FROM RankingIntercambios WH
```

ATAPUERCA NET – CONFIDENCIAL

```
ERE RankingBeneficio <= 2 UNION ALL SELECT 'RESUMEN ECONÓMICO' AS BaseOrigen, 'Sistema de
intercambio' AS BaseDestino, SUM(ComidaATransferir) AS TotalComidaIntercambio, SUM(AguaA
Transferir) AS TotalAguaIntercambio, AVG(PotencialBeneficio) AS BeneficioPromedio, CASE W
HEN COUNT(*) >= 4 THEN '■ Economía dinámica posible' WHEN COUNT(*) >= 2 THEN '■ Intercamb
ios limitados' ELSE '■ Economía estancada' END AS EstadoEconomico, CASE WHEN AVG(Potencia
lBeneficio) > 75 THEN '■ Mercado eficiente' ELSE '■ Optimización necesaria' END AS Eficie
nciaMercado FROM RankingIntercambios ORDER BY PotencialBeneficio DESC;
```

MISIÓN 58: ■■ EVALUACIÓN DE LIDERAZGO Y SUCESIÓN

****Fecha:**** 8 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** C
omando Central – General Luna

****Contexto narrativo****

Operación maestra. CTEs encadenadas y simulaciones para decidir el futuro del
asentamiento.

****Objetivo****

Tres CTEs: LiderazgoActual (puntuación por rol), CandidatosSucesion (CROSS JOIN evalúa
gaps), AnalisisCriticidad (GROUP BY rol analiza disponibilidad). Identifica mejores
candidatos por mínima diferencia liderazgo.

****Notas de inteligencia****

Análisis de sucesión ultra-complejo: evalúa liderazgo actual, identifica candidatos,
analiza gaps generacionales y riesgos de continuidad

****Consulta sugerida****

[SQL]

```
WITH LiderazgoActual AS (SELECT s.SurvivorID, s.Nombre, s.Edad, s.Rol, b.Nombre AS Base,
b.EsComandoCentral, CASE WHEN s.Rol = 'Comandante' THEN 100 WHEN s.Rol = 'Médica' THEN 90
WHEN s.Rol = 'Soldado' THEN 70 WHEN s.Rol = 'Ingeniera' THEN 75 ELSE 50 END AS Puntuacio
nLiderazgo FROM Survivors s JOIN Bases b ON s.BaseID = b.BaseID), CandidatosSucesion AS (
SELECT la.Nombre AS LiderActual, la.Edad AS EdadLider, la.Rol AS RolLider, la.Base, s2.No
mbre AS PosibleSucesor, s2.Edad AS EdadSucesor, s2.Rol AS RolSucesor, ABS(la.PuntuacionLi
derazgo - CASE WHEN s2.Rol = 'Comandante' THEN 100 WHEN s2.Rol = 'Médica' THEN 90 WHEN s2
.Rol = 'Soldado' THEN 70 WHEN s2.Rol = 'Ingeniera' THEN 75 ELSE 50 END) AS DiferenciaLide
razgo, CASE WHEN la.Edad - s2.Edad > 5 THEN 'Sucesión generacional saludable' WHEN ABS(la
.Edad - s2.Edad) <= 5 THEN 'Sucesión entre pares' ELSE 'Líder más joven que candidato' EN
D AS TipoSucesion FROM LiderazgoActual la CROSS JOIN Survivors s2 WHERE la.SurvivorID !=
s2.SurvivorID AND la.PuntuacionLiderazgo >= 70), AnalisisCriticidad AS (SELECT RolLider,
COUNT(DISTINCT LiderActual) AS LideresActuales, COUNT(DISTINCT PosibleSucesor) AS Candida
tosDisponibles, AVG(EdadLider) AS EdadPromedioLideres, MIN(DiferenciaLiderazgo) AS MejorC
andidatoGap, CASE WHEN COUNT(DISTINCT LiderActual) = 1 AND COUNT(DISTINCT PosibleSucesor)
= 0 THEN '■ CRISIS - Sin sucesión' WHEN COUNT(DISTINCT LiderActual) = 1 AND COUNT(DISTIN
CT PosibleSucesor) < 2 THEN '■ RIESGO - Sucesión limitada' WHEN COUNT(DISTINCT LiderActua
l) = 1 THEN '■ ESTABLE - Sucesión disponible' ELSE '■ ROBUSTO - Múltiple liderazgo' END A
S EstadoSucesion FROM CandidatosSucesion GROUP BY RolLider) SELECT LiderActual, EdadLider
, RolLider, Base, PosibleSucesor, EdadSucesor, RolSucesor, DiferenciaLiderazgo, TipoSuces
```

```
ion, CASE WHEN DiferenciaLiderazgo <= 10 THEN '■ Candidato ideal' WHEN DiferenciaLiderazgo <= 20 THEN '■ Candidato viable' ELSE '■ Candidato subóptimo' END AS CalidadCandidato FROM CandidatosSucesion WHERE DiferenciaLiderazgo = (SELECT MIN(cs2.DiferenciaLiderazgo) FROM CandidatosSucesion cs2 WHERE cs2.LiderActual = CandidatosSucesion.LiderActual) UNION ALL SELECT 'ANÁLISIS GLOBAL' AS LiderActual, NULL AS EdadLider, RolLider, 'Todo el asentamiento' AS Base, NULL AS PosibleSucesor, NULL AS EdadSucesor, NULL AS RolSucesor, NULL AS DiferenciaLiderazgo, EstadoSucesion AS TipoSucesion, CASE WHEN EstadoSucesion LIKE '%CRISIS%' THEN '■ Intervención urgente' WHEN EstadoSucesion LIKE '%RIESGO%' THEN '■ Planificación necesaria' ELSE '■ Situación controlada' END AS CalidadCandidato FROM AnalisisCriticidad ORDER BY CASE WHEN LiderActual = 'ANÁLISIS GLOBAL' THEN 1 ELSE 0 END, DiferenciaLiderazgo;
```

MISIÓN 59: ■ MODELADO DE EXPANSIÓN TERRITORIAL

****Fecha:**** 9 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación maestra. CTes encadenadas y simulaciones para decidir el futuro del asentamiento.

****Objetivo****

Cuatro CTes: BasesDatos (recursos totales), ZonasExpansion (UNION ALL genera coordenadas +1/-1), EvaluacionExpansion (algoritmo viabilidad: población×20+recursos/10-distancia×15), PlanExpansionGlobal (estadísticas). UNION ALL plan maestro.

****Notas de inteligencia****

Modelado territorial mega-complejo: evalúa múltiples zonas de expansión, calcula viabilidad basada en población+recursos-distancia, prioriza por potencial estratégico

****Consulta sugerida****

[SQL]

```
WITH BasesDatos AS (SELECT b.BaseID, b.Nombre AS Base, b.Latitud, b.Longitud, b.TipoBase, b.Capacidad, COUNT(s.SurvivorID) AS PoblacionActual, COALESCE(r.ComidaRaciones + r.AguaLitros + r.Armas + r.Medicinas, 0) AS RecursosTotales FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID GROUP BY b.BaseID, b.Nombre, b.Latitud, b.Longitud, b.TipoBase, b.Capacidad, r.ComidaRaciones, r.AguaLitros, r.Armas, r.Medicinas), ZonasExpansion AS (SELECT bd.Base AS BaseReferencia, bd.Latitud + 1 AS NuevaLatitud, bd.Longitud + 1 AS NuevaLongitud, 'Noreste de ' + bd.Base AS NombreZona, bd.PoblacionActual, bd.RecursosTotales, SQRT(POWER(bd.Latitud + 1 - (SELECT AVG(Latitud) FROM BasesDatos), 2) + POWER(bd.Longitud + 1 - (SELECT AVG(Longitud) FROM BasesDatos), 2)) AS DistanciaAlCentroRed FROM BasesDatos bd WHERE bd.TipoBase = 'Humana' AND bd.PoblacionActual > 0 UNION ALL SELECT bd.Base AS BaseReferencia, bd.Latitud - 1 AS NuevaLatitud, bd.Longitud - 1 AS NuevaLongitud, 'Suroeste de ' + bd.Base AS NombreZona, bd.PoblacionActual, bd.RecursosTotales, SQRT(POWER(bd.Latitud - 1 - (SELECT AVG(Latitud) FROM BasesDatos), 2) + POWER(bd.Longitud - 1 - (SELECT AVG(Longitud) FROM BasesDatos), 2)) AS DistanciaAlCentroRed FROM BasesDatos bd WHERE bd.TipoBase = 'Humana' AND bd.PoblacionActual > 0), E
```

```

valuacionExpansion AS (SELECT BaseReferencia, NombreZona, NuevaLatitud, NuevaLongitud, Po
blacionActual, RecursosTotales, ROUND(DistanciaAlCentroRed, 4) AS DistanciaAlCentro, (Pob
lacionActual * 20 + RecursosTotales / 10.0 - DistanciaAlCentroRed * 15) AS PuntuacionViab
ilidad, CASE WHEN (PoblacionActual * 20 + RecursosTotales / 10.0 - DistanciaAlCentroRed *
15) > 100 THEN '■ Expansión altamente viable' WHEN (PoblacionActual * 20 + RecursosTotal
es / 10.0 - DistanciaAlCentroRed * 15) > 50 THEN '■ Expansión viable con riesgos' WHEN (P
oblacionActual * 20 + RecursosTotales / 10.0 - DistanciaAlCentroRed * 15) > 0 THEN '■ Exp
ansión arriesgada' ELSE '■ Expansión no recomendada' END AS RecomendacionExpansion, CASE
WHEN DistanciaAlCentro < 2 THEN 'Zona central estratégica' WHEN DistanciaAlCentro < 4 THE
N 'Zona intermedia' ELSE 'Zona periférica' END AS ClasificacionEstrategica FROM ZonasExpa
nsion), PlanExpansionGlobal AS (SELECT COUNT(*) AS ZonasEvaluadas, COUNT(CASE WHEN Recome
ndacionExpansion LIKE '%viable%' THEN 1 END) AS ZonasViabiles, AVG(PuntuacionViabilidad) A
S ViabilidadPromedio, MAX(PuntuacionViabilidad) AS MejorViabilidad, MIN(PuntuacionViabili
dad) AS PeorViabilidad FROM EvaluacionExpansion) SELECT BaseReferencia, NombreZona, Nueva
Latitud, NuevaLongitud, ROUND(PuntuacionViabilidad, 2) AS PuntuacionViabilidad, Recomena
cionExpansion, ClasificacionEstrategica, CASE WHEN PuntuacionViabilidad = (SELECT MejorVi
abilidad FROM PlanExpansionGlobal) THEN '■ PRIORIDAD MÁXIMA' WHEN PuntuacionViabilidad >
(SELECT ViabilidadPromedio FROM PlanExpansionGlobal) THEN '■ Alta prioridad' ELSE '■ Cons
iderar más adelante' END AS PrioridadExpansion FROM EvaluacionExpansion UNION ALL SELECT
'PLAN MAESTRO' AS BaseReferencia, 'Expansión territorial global' AS NombreZona, NULL AS N
uevaLatitud, NULL AS NuevaLongitud, ViabilidadPromedio AS PuntuacionViabilidad, CASE WHEN
(ZonasViabiles * 100.0 / ZonasEvaluadas) > 50 THEN '■ Expansión territorial recomendada'
WHEN (ZonasViabiles * 100.0 / ZonasEvaluadas) > 25 THEN '■ Expansión selectiva recomendada
' ELSE '■ Consolidación antes que expansión' END AS RecomendacionExpansion, CAST(ZonasVia
bles AS VARCHAR) + '/' + CAST(ZonasEvaluadas AS VARCHAR) + ' zonas viables (' + CAST(ROUN
D(ZonasViabiles * 100.0 / ZonasEvaluadas, 1) AS VARCHAR) + '%)' AS ClasificacionEstrategic
a, CASE WHEN MejorViabilidad > 100 THEN '■ Oportunidades excelentes disponibles' ELSE '■
Expansión conservadora recomendada' END AS PrioridadExpansion FROM PlanExpansionGlobal OR
DER BY PuntuacionViabilidad DESC;

```

MISIÓN 60: ■ GRAN SIMULACIÓN: EL FUTURO DE ATAPUERCA

Fecha: 10 de octubre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:**
Comando Central – General Luna

Contexto narrativo

Operación maestra. CTEs encadenadas y simulaciones para decidir el futuro del asentamiento.

Objetivo

Cinco CTEs complejos: BaselineActual (estado actual), ProyeccionDemografica (crecimiento poblacional), ProyeccionRecursos (sostenibilidad), AnalisisLiderazgo (continuidad), EscenariosFuturos (predicciones). CROSS JOIN múltiple integra todos los factores para proyectar 5 años.

Notas de inteligencia

La consulta más compleja del sistema: integra demografía, recursos, liderazgo para proyectar escenarios futuros a 5 años con estrategias específicas

****Consulta sugerida****

[SQL]

```
WITH BaselineActual AS (SELECT COUNT(DISTINCT b.BaseID) AS BasesTotales, COUNT(DISTINCT s
.SurvivorID) AS SupervivientesTotales, SUM(COALESCE(r.ComidaRaciones + r.AguaLitros + r.Armas + r.Medicinas, 0)) AS RecursosTotales, AVG(CAST(s.Edad AS FLOAT)) AS EdadPromedio, COUNT(CASE WHEN s.Rol IN ('Comandante', 'Médica') THEN 1 END) AS LiderazgoTotal FROM Bases
b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID), ProyeccionDemografica AS (SELECT ba.SupervivientesTotales, ba.EdadPromedio, ba.LiderazgoTotal, CASE WHEN ba.EdadPromedio < 30 THEN ba.SupervivientesTotales * 1.3 WHEN ba.EdadPromedio < 40 THEN ba.SupervivientesTotales * 1.1 ELSE ba.SupervivientesTotales * 0.9 END AS PoblacionProyectada5Anos, CASE WHEN ba.EdadPromedio < 30 THEN 'Crecimiento demográfico' WHEN ba.EdadPromedio < 40 THEN 'Estabilidad demográfica' ELSE 'Declive demográfico' END AS TendenciaDemografica FROM BaselineActual ba), ProyeccionRecursos AS (SELECT ba.RecursosTotales, ba.SupervivientesTotales, CASE WHEN ba.RecursosTotales / CAST(ba.SupervivientesTotales AS FLOAT) > 200 THEN ba.RecursosTotales * 1.2 WHEN ba.RecursosTotales / CAST(ba.SupervivientesTotales AS FLOAT) > 100 THEN ba.RecursosTotales * 1.0 ELSE ba.RecursosTotales * 0.8 END AS RecursosProyectados5Anos, CASE WHEN ba.RecursosTotales / CAST(ba.SupervivientesTotales AS FLOAT) > 200 THEN 'Abundancia sostenible' WHEN ba.RecursosTotales / CAST(ba.SupervivientesTotales AS FLOAT) > 100 THEN 'Equilibrio precario' ELSE 'Escasez creciente' END AS TendenciaRecursos FROM BaselineActual ba), AnalisisLiderazgo AS (SELECT ba.LiderazgoTotal, ba.SupervivientesTotales, CASE WHEN ba.LiderazgoTotal / CAST(ba.SupervivientesTotales AS FLOAT) > 0.3 THEN 'Liderazgo sólido' WHEN ba.LiderazgoTotal / CAST(ba.SupervivientesTotales AS FLOAT) > 0.1 THEN 'Liderazgo suficiente' ELSE 'Crisis de liderazgo' END AS EstadoLiderazgo, CASE WHEN ba.LiderazgoTotal >= 2 THEN ba.LiderazgoTotal + 1 ELSE ba.LiderazgoTotal END AS LiderazgoProyectado5Anos FROM BaselineActual ba), EscenariosFuturos AS (SELECT pd.PoblacionProyectada5Anos, pr.RecursosProyectados5Anos, al.LiderazgoProyectado5Anos, pd.TendenciaDemografica, pr.TendenciaRecursos, al.EstadoLiderazgo, CASE WHEN pd.TendenciaDemografica = 'Crecimiento demográfico' AND pr.TendenciaRecursos = 'Abundancia sostenible' AND al.EstadoLiderazgo = 'Liderazgo sólido' THEN '■ GOLDEN AGE - Prosperidad total' WHEN pd.TendenciaDemografica != 'Declive demográfico' AND pr.TendenciaRecursos != 'Escasez creciente' AND al.EstadoLiderazgo != 'Crisis de liderazgo' THEN '■ ESTABILIDAD - Supervivencia asegurada' WHEN pd.TendenciaDemografica = 'Declive demográfico' OR pr.TendenciaRecursos = 'Escasez creciente' OR al.EstadoLiderazgo = 'Crisis de liderazgo' THEN '■ DESAFÍOS - Supervivencia incierta' ELSE '■ COLAPSO - Extinción probable' END AS EscenarioFuturo, (pd.PoblacionProyectada5Anos * 10 + pr.RecursosProyectados5Anos / 50.0 + al.LiderazgoProyectado5Anos * 100) AS PuntuacionSupervivencia5Anos FROM ProyeccionDemografica pd CROSS JOIN ProyeccionRecursos pr CROSS JOIN AnalisisLiderazgo al), RecomendacionesEstrategicas AS (SELECT ef.EscenarioFuturo, ef.PuntuacionSupervivencia5Anos, CASE WHEN ef.EscenarioFuturo LIKE '%GOLDEN AGE%' THEN 'Expandir territorio, establecer nuevas bases, liderar región' WHEN ef.EscenarioFuturo LIKE '%ESTABILIDAD%' THEN 'Consolidar recursos, formar alianzas, crecimiento moderado' WHEN ef.EscenarioFuturo LIKE '%DESAFÍOS%' THEN 'Modo supervivencia, conservar recursos, fortalecer defensas' ELSE 'Evacuar población, buscar refugio, medidas desesperadas' END AS EstrategiaRecomendada, CASE WHEN ef.PuntuacionSupervivencia5Anos > 1000 THEN 'Civilización próspera' WHEN ef.PuntuacionSupervivencia5Anos > 500 THEN 'Asentamiento estable' WHEN ef.PuntuacionSupervivencia5Anos > 200 THEN 'Supervivencia básica' ELSE 'Extinción inminente' END AS DestinoFinal FROM EscenariosFuturos ef) SELECT 'PROYECCIÓN 2030' AS Categoria, 'Población: ' + CAST(ROUND(pd.PoblacionProyectada5Anos, 0) AS VARCHAR) + ' | Recursos: ' + CAST(ROUND(pr.RecursosProyectados5Anos, 0) AS VARCHAR) + ' | Líderes: ' + CAST(al.LiderazgoProyectado5Anos AS VARCHAR) AS DatosProyectados
```



```
, ef.EscenarioFuturo AS Evaluacion, re.EstrategiaRecomendada AS AccionRequerida, re.DestinoFinal AS ResultadoFinal FROM ProyeccionDemografica pd CROSS JOIN ProyeccionRecursos pr CROSS JOIN AnalisisLiderazgo al CROSS JOIN EscenariosFuturos ef CROSS JOIN RecomendacionesEstrategicas re UNION ALL SELECT 'ESTADO ACTUAL' AS Categoria, 'Población: ' + CAST(ba.SuperVivientesTotales AS VARCHAR) + ' | Recursos: ' + CAST(ba.RecursosTotales AS VARCHAR) + ' | Líderes: ' + CAST(ba.LiderazgoTotal AS VARCHAR) AS DatosProyectados, pd.TendenciaDemografica + ' | ' + pr.TendenciaRecursos + ' | ' + al.EstadoLiderazgo AS Evaluacion, 'Análisis base para proyecciones futuras' AS AccionRequerida, 'Punto de partida para simulación' AS ResultadoFinal FROM BaselineActual ba CROSS JOIN ProyeccionDemografica pd CROSS JOIN ProyeccionRecursos pr CROSS JOIN AnalisisLiderazgo al ORDER BY CASE WHEN Categoria = 'ESTADO ACTUAL' THEN 1 ELSE 2 END;
```

MISIÓN 61: ■ CATÁLOGO DE AMENAZAS ROBÓTICAS

****Fecha:**** 11 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que sea en nuestro tablero.

****Objetivo****

De la tabla Robots, selecciona modelo, generación, funciones y nivel amenaza. Usa CASE WHEN para clasificar riesgo según nivel: ≥9 crítico, ≥7 alto, ≥5 medio, <5 bajo. Ordena por amenaza y generación descendente.

****Notas de inteligencia****

Usa la tabla Robots. Analiza el nivel de amenaza para clasificar el riesgo. Los robots más modernos suelen ser más peligrosos

****Consulta sugerida****

[SQL]

```
SELECT r.Modelo, r.Generacion, r.Funciones, r.NivelAmenaza, CASE WHEN r.NivelAmenaza >= 9 THEN '■ CRÍTICO' WHEN r.NivelAmenaza >= 7 THEN '■ ALTO' WHEN r.NivelAmenaza >= 5 THEN '■ MEDIO' ELSE '■ BAJO' END AS ClasificacionRiesgo FROM Robots r ORDER BY r.NivelAmenaza DESC, r.Generacion DESC;
```

MISIÓN 62: ■ HISTORIAL DE ATAQUES POR BASE

****Fecha:**** 12 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Aplicación real en campo. Conecta piezas y valida hipótesis: si la IA cruza señales, que sea en nuestro tablero.

****Objetivo****

LEFT JOIN de Bases (b) con Attacks (a), agrupa por base. Cuenta ataques con COUNT, suma muertos con SUM, última fecha con MAX y concatena tipos robot con STRING_AGG. Ordena por ataques y muertes descendente.

****Notas de inteligencia****

Usa JOIN entre Bases y Attacks. STRING_AGG concatena los tipos de robots. Ordena por mayor cantidad de ataques

****Consulta sugerida****

[SQL]

```
SELECT b.Nombre AS Base, COUNT(a.AttackID) AS TotalAtaques, SUM(a.Muertos) AS TotalMuertos,
MAX(a.Fecha) AS UltimoAtaque, STRING_AGG(a.TipoRobot, ', ') AS RobotsAgresores FROM Bases b
LEFT JOIN Attacks a ON b.BaseID = a.BaseID GROUP BY b.BaseID, b.Nombre ORDER BY TotalAtaques DESC, TotalMuertos DESC;
```

MISIÓN 63: ■ EFICIENCIA DE MISIONES POR ALIANZA

****Fecha:**** 13 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

****Objetivo****

De Missions (m), usa CASE WHEN con EXISTS para verificar si origen y destino están aliados (consulta Alliances bidireccionalmente). Agrupa por tipo cooperación, calcula porcentaje éxito dividiendo completadas entre total.

****Notas de inteligencia****

Usa EXISTS para verificar alianzas entre origen y destino. Calcula porcentaje de éxito comparando estados 'Completada'

****Consulta sugerida****

[SQL]

```
SELECT CASE WHEN EXISTS (SELECT 1 FROM Alliances al WHERE (al.BaseID1 = m.OrigenID AND al.BaseID2 = m.DestinoID) OR (al.BaseID1 = m.DestinoID AND al.BaseID2 = m.OrigenID)) THEN 'Misión entre Aliados' ELSE 'Misión Independiente' END AS TipoCooperacion, COUNT(*) AS TotalMisiones, SUM(CASE WHEN Estado = 'Completada' THEN 1 ELSE 0 END) AS MisionesExitosas, ROUND(SUM(CASE WHEN Estado = 'Completada' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) AS PorcentajeExito FROM Missions m GROUP BY CASE WHEN EXISTS (SELECT 1 FROM Alliances al WHERE (al.BaseID1 = m.OrigenID AND al.BaseID2 = m.DestinoID) OR (al.BaseID1 = m.DestinoID AND al.BaseID2 = m.OrigenID)) THEN 'Misión entre Aliados' ELSE 'Misión Independiente' END;
```


MISIÓN 64: ■■ CORRELACIÓN AVISTAMIENTOS-ATAQUES

****Fecha:**** 14 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
Comando Central – General Luna

****Contexto narrativo****

Inteligencia operativa. Las cifras hablan: promedios, rankings y densidades para anticipar movimientos.

****Objetivo****

CTE AvistamientosAmenaza filtra RobotSightings WHERE NivelAmenaza≥7. JOIN con Attacks usando DATEDIFF para calcular días entre avistamiento y ataque. Filtra WHERE ataque≥avistamiento Y diferencia≤10 días. Clasifica patrón temporal.

****Notas de inteligencia****

CTE para filtrar avistamientos peligrosos, luego JOIN con ataques. DATEDIFF calcula días entre eventos

****Consulta sugerida****

[SQL]

```
WITH AvistamientosAmenaza AS (SELECT rs.BaseID, rs.Fecha AS FechaAvistamiento, rs.TipoRobot, rs.NivelAmenaza, b.Nombre AS Base FROM RobotSightings rs JOIN Bases b ON rs.BaseID = b.BaseID WHERE rs.NivelAmenaza >= 7) SELECT av.Base, av.FechaAvistamiento, av.TipoRobot AS RobotAvistado, av.NivelAmenaza, a.Fecha AS FechaAtaque, a.TipoRobot AS RobotAtacante, a.Muertos, DATEDIFF(day, av.FechaAvistamiento, a.Fecha) AS DiasDeWarning, CASE WHEN DATEDIFF(day, av.FechaAvistamiento, a.Fecha) BETWEEN 0 AND 3 THEN '■ Ataque Inmediato' WHEN DATEDIFF(day, av.FechaAvistamiento, a.Fecha) BETWEEN 4 AND 7 THEN '■ Ataque Próximo' ELSE '■ Sin Correlación Directa' END AS PatronTemporal FROM AvistamientosAmenaza av JOIN Attacks a ON av.BaseID = a.BaseID WHERE a.Fecha >= av.FechaAvistamiento AND DATEDIFF(day, av.FechaAvistamiento, a.Fecha) <= 10 ORDER BY DiasDeWarning, av.NivelAmenaza DESC;
```

MISIÓN 65: ■ RED DE SUMINISTROS INTELIGENTE

****Fecha:**** 15 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
Comando Central – General Luna

****Contexto narrativo****

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

****Objetivo****

Dos CTEs: NecesidadRecursos (calcula necesidades: personal×100 comida, ×20 medicinas) y SuministrosRecientes (suma entregas desde julio 2025). LEFT JOIN entre CTEs, compara necesidades vs disponible para detectar déficit.

****Notas de inteligencia****

Dos CTEs: una calcula necesidades basadas en personal, otra suma suministros recientes. Compara deficit vs entregas

****Consulta sugerida****

[SQL]

```
WITH NecesidadRecursos AS (SELECT b.BaseID, b.Nombre AS Base, COUNT(s.SurvivorID) AS Personal, COALESCE(r.ComidaRaciones, 0) AS ComidaActual, COALESCE(r.Medicinas, 0) AS MedicinasActuales, COUNT(s.SurvivorID) * 100 AS ComidaNecesaria, COUNT(s.SurvivorID) * 20 AS MedicinasNecesarias FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID GROUP BY b.BaseID, b.Nombre, r.ComidaRaciones, r.Medicinas), SuministrosRecientes AS (SELECT su.BaseID, su.Tipo, SUM(su.Cantidad) AS CantidadEntregada, MAX(su.FechaEntrega) AS UltimaEntrega FROM Supplies su WHERE su.FechaEntrega >= '2025-07-01' GROUP BY su.BaseID, su.Tipo) SELECT nr.Base, nr.Personal, nr.ComidaActual, nr.ComidaNecesaria, (nr.ComidaActual - nr.ComidaNecesaria) AS DeficitComida, nr.MedicinasActuales, nr.MedicinasNecesarias, (nr.MedicinasActuales - nr.MedicinasNecesarias) AS DeficitMedicinas, COALESCE(src.CantidadEntregada, 0) AS SuministrosRecibidos, src.UltimaEntrega, CASE WHEN (nr.ComidaActual - nr.ComidaNecesaria) < -50 OR (nr.MedicinasActuales - nr.MedicinasNecesarias) < -10 THEN '■ CRISIS - Necesita suministros urgentes' WHEN (nr.ComidaActual - nr.ComidaNecesaria) < 0 OR (nr.MedicinasActuales - nr.MedicinasNecesarias) < 0 THEN '■ RIESGO - Suministros insuficientes' ELSE '■ ESTABLE - Suministros adecuados' END AS EstadoLogistico FROM NecesidadRecursos nr LEFT JOIN SuministrosRecientes src ON nr.BaseID = src.BaseID ORDER BY DeficitComida, DeficitMedicinas;
```

MISIÓN 66: ■■ ANÁLISIS DE RUTAS ESTRATÉGICAS

****Fecha:**** 16 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:**** Comando Central – General Luna

****Contexto narrativo****

Operación de riesgo. Subconsultas y filtros quirúrgicos para detectar vulnerabilidades y ventajas.

****Objetivo****

JOIN de DistanceMatrix (dm) con Bases origen (bo) y destino (bd), LEFT JOIN con Missions (m). Clasifica rutas por kilómetros, cuenta misiones realizadas, calcula porcentaje éxito dividiendo completadas/total. Ordena por uso y distancia.

****Notas de inteligencia****

Une DistanceMatrix con Bases (dos veces) y Missions. Calcula éxito por ruta y clasifica distancias

****Consulta sugerida****

[SQL]

```
SELECT bo.Nombre AS Origen, bd.Nombre AS Destino, dm.Kilometros, CASE WHEN dm.Kilometros <= 50 THEN '■ Corta' WHEN dm.Kilometros <= 100 THEN '■ Media' ELSE '■ Larga' END AS TipoRuta, COUNT(m.MissionID) AS MisionesRealizadas, CASE WHEN COUNT(m.MissionID) > 0 THEN 'Rut
```

ATAPUERCA NET – CONFIDENCIAL

```
a Activa' ELSE 'Ruta Potencial' END AS EstadoUso, CASE WHEN COUNT(m.MissionID) > 0 THEN R
OUND(SUM(CASE WHEN m.Estado = 'Completada' THEN 1 ELSE 0 END) * 100.0 / COUNT(m.MissionID
), 2) ELSE 0 END AS PorcentajeExitoRuta FROM DistanceMatrix dm JOIN Bases bo ON dm.IDOrig
en = bo.BaseID JOIN Bases bd ON dm.IDDestino = bd.BaseID LEFT JOIN Missions m ON dm.IDOri
gen = m.OrigenID AND dm.IDDestino = m.DestinoID GROUP BY bo.BaseID, bo.Nombre, bd.BaseID,
bd.Nombre, dm.Kilometros ORDER BY MisionesRealizadas DESC, dm.Kilometros;
```

MISIÓN 67: ■ ÍNDICE DE SEGURIDAD INTEGRAL

****Fecha:**** 17 de octubre de 2036 | ****Clasificación:**** ULTRASECRETO | ****Emisor:****
Comando Central – General Luna

****Contexto narrativo****

Operación maestra. CTes encadenadas y simulaciones para decidir el futuro del asentamiento.

****Objetivo****

CTE BaseSeguridad: LEFT JOIN múltiple de Bases con Survivors, Attacks, RobotSightings, Alliances, Resources. Cuenta eventos por base, calcula índice con algoritmo: (+personal×15 +alianzas×20 +armas/10) - (ataques×15 +avistamientos×5 +amenaza×2). Limita 0-100.

****Notas de inteligencia****

CTE que une 6 tablas para calcular factores de seguridad. Fórmula: suma factores positivos, resta amenazas

****Consulta sugerida****

[SQL]

```
WITH BaseSeguridad AS (SELECT b.BaseID, b.Nombre AS Base, COUNT(DISTINCT s.SurvivorID) AS
Personal, COUNT(DISTINCT a.AttackID) AS AtaquesRecibidos, COUNT(DISTINCT rs.SightingID)
AS AvistamientosRobots, COUNT(DISTINCT al.AllianceID) AS AlianzasActivas, COALESCE(r.Armas, 0) AS Armamento, MAX(a.Fecha) AS UltimoAtaque, MAX(rs.NivelAmenaza) AS MaxAmenazaDetectada FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Attacks a ON b.BaseID = a.BaseID LEFT JOIN RobotSightings rs ON b.BaseID = rs.BaseID LEFT JOIN Alliances al ON b.BaseID = al.BaseID1 OR b.BaseID = al.BaseID2 LEFT JOIN Resources r ON b.BaseID = r.BaseID GROUP BY b.BaseID, b.Nombre, r.Armas) SELECT Base, Personal, AtaquesRecibidos, AvistamientosRobots, AlianzasActivas, Armamento, UltimoAtaque, COALESCE(MaxAmenazaDetectada, 0) AS AmenazaMaxima, ROUND(GREATEST(0, LEAST(100, (Personal * 15) + (AlianzasActivas * 20) + (Armamento / 10.0) - (AtaquesRecibidos * 15) - (AvistamientosRobots * 5) - (COALESCE(MaxAmenazaDetectada, 0) * 2))), 2) AS IndiceSeguridadIntegral, CASE WHEN ROUND(GREATEST(0, LEAST(100, (Personal * 15) + (AlianzasActivas * 20) + (Armamento / 10.0) - (AtaquesRecibidos * 15) - (AvistamientosRobots * 5) - (COALESCE(MaxAmenazaDetectada, 0) * 2))), 2) >= 80 THEN '■ FORTALEZA SEGURA' WHEN ROUND(GREATEST(0, LEAST(100, (Personal * 15) + (AlianzasActivas * 20) + (Armamento / 10.0) - (AtaquesRecibidos * 15) - (AvistamientosRobots * 5) - (COALESCE(MaxAmenazaDetectada, 0) * 2))), 2) >= 60 THEN '■ DEFENSA ACEPTABLE' WHEN ROUND(GREATEST(0, LEAST(100, (Personal * 15) + (AlianzasActivas * 20) + (Armamento / 10.0) - (AtaquesRecibidos * 15) - (AvistamientosRobots * 5) - (COALESCE(MaxAmenazaDetectada, 0) * 2))), 2) >= 40 THEN '■ EN RIESGO' ELSE '■ CRÍTICO' END AS ClasificacionSeguridad FR
```

MISIÓN 68: ■ MATRIZ DE PODER E INFLUENCIA

Fecha: 18 de octubre de 2036 | **Clasificación:** ULTRASECRETO | **Emisor:** Comando Central – General Luna

****Contexto narrativo****

Operación maestra. CTEs encadenadas y simulaciones para decidir el futuro del asentamiento.

****Objetivo****

Tres CTEs: PoderMilitar (personal, comandantes, armas, suministros), InfluenciaEstrategica (alianzas, misiones, confianza), VulnerabilidadOperacional (ataques, avistamientos, distancias). JOIN entre CTEs, algoritmo poder integral con múltiples factores ponderados.

****Notas de inteligencia****

Tres CTEs para diferentes aspectos del poder. Fórmula integral que suma factores positivos y resta vulnerabilidades

****Consulta sugerida****

[SQL]

WITH PoderMilitar AS (SELECT b.BaseID, b.Nombre AS Base, COUNT(DISTINCT s.SurvivorID) AS Personal, COUNT(CASE WHEN s.Rol = 'Comandante' THEN 1 END) AS Comandantes, COALESCE(r.Armas, 0) AS Armamento, COUNT(DISTINCT su.SupplyID) AS SuministrosRecibidos FROM Bases b LEFT JOIN Survivors s ON b.BaseID = s.BaseID LEFT JOIN Resources r ON b.BaseID = r.BaseID LEFT JOIN Supplies su ON b.BaseID = su.BaseID GROUP BY b.BaseID, b.Nombre, r.Armas), Influen-
ciaEstrategica AS (SELECT b.BaseID, COUNT(DISTINCT al.AllianceID) AS AlianzasActivas, CO-
UNT(DISTINCT m.MissionID) AS MisionesLideradas, AVG(CAST(al.NivelConfianza AS FLOAT)) AS
ConfianzaPromedio FROM Bases b LEFT JOIN Alliances al ON b.BaseID = al.BaseID1 OR b.BaseI-
D = al.BaseID2 LEFT JOIN Missions m ON b.BaseID = m.OrigenID GROUP BY b.BaseID), Vulnerab-
ilidadOperacional AS (SELECT b.BaseID, COUNT(DISTINCT a.AttackID) AS AtaquesRecibidos, CO-
UNT(DISTINCT rs.SightingID) AS AvistamientosRobots, MAX(a.Muertos) AS MaxBajasEnAtaque, A-
VG(CAST(dm.Kilometros AS FLOAT)) AS DistanciaPromedioAOtrasBases FROM Bases b LEFT JOIN A-
ttacks a ON b.BaseID = a.BaseID LEFT JOIN RobotSightings rs ON b.BaseID = rs.BaseID LEFT
JOIN DistanceMatrix dm ON b.BaseID = dm.IDOrigen GROUP BY b.BaseID) SELECT pm.Base, pm.Pe-
rsonal, pm.Comandantes, pm.Armamento, ie.AlianzasActivas, ie.MisionesLideradas, COALESCE(
ie.ConfianzaPromedio, 0) AS ConfianzaPromedio, vo.AtaquesRecibidos, vo.AvistamientosRobot-
s, COALESCE(vo.MaxBajasEnAtaque, 0) AS MaxBajas, ROUND(COALESCE(vo.DistanciaPromedioAOtra-
sBases, 0), 2) AS DistanciaPromedio, ROUND((pm.Personal * 30) + (pm.Comandantes * 50) + (
pm.Armamento / 2.0) + (ie.AlianzasActivas * 40) + (ie.MisionesLideradas * 25) + (COALESCE(
ie.ConfianzaPromedio, 0) * 10) + (pm.SuministrosRecibidos * 15) - (vo.AtaquesRecibidos *
20) - (vo.AvistamientosRobots * 10) - (COALESCE(vo.MaxBajasEnAtaque, 0) * 30), 2) AS Ind-
icePoder, CASE WHEN ((pm.Personal * 30) + (pm.Comandantes * 50) + (pm.Armamento / 2.0) +
(ie.AlianzasActivas * 40) + (ie.MisionesLideradas * 25) + (COALESCE(ie.ConfianzaPromedio,
0) * 10) + (pm.SuministrosRecibidos * 15) - (vo.AtaquesRecibidos * 20) - (vo.Avistamient

ATAPUERCA NET – CONFIDENCIAL

```
osRobots * 10) - (COALESCE(vo.MaxBajasEnAtaque, 0) * 30)) >= 500 THEN '■ POTENCIA HEGEMÓNICA' WHEN ((pm.Personal * 30) + (pm.Comandantes * 50) + (pm.Armamento / 2.0) + (ie.AlianzasActivas * 40) + (ie.MisionesLideradas * 25) + (COALESCE(ie.ConfianzaPromedio, 0) * 10) + (pm.SuministrosRecibidos * 15) - (vo.AtaquesRecibidos * 20) - (vo.AvistamientosRobots * 10) - (COALESCE(vo.MaxBajasEnAtaque, 0) * 30)) >= 300 THEN '■ POTENCIA REGIONAL' WHEN ((pm.Personal * 30) + (pm.Comandantes * 50) + (pm.Armamento / 2.0) + (ie.AlianzasActivas * 40) + (ie.MisionesLideradas * 25) + (COALESCE(ie.ConfianzaPromedio, 0) * 10) + (pm.SuministrosRecibidos * 15) - (vo.AtaquesRecibidos * 20) - (vo.AvistamientosRobots * 10) - (COALESCE(vo.MaxBajasEnAtaque, 0) * 30)) >= 100 THEN '■■■ ACTOR RELEVANTE' ELSE '■ BASE IRRELEVANTE' END AS ClasificacionPoder FROM PoderMilitar pm JOIN InfluenciaEstrategica ie ON pm.BaseID = ie.BaseID JOIN VulnerabilidadOperacional vo ON pm.BaseID = vo.BaseID ORDER BY IndicePoder DESC;
```

EPÍLOGO

Si este dossier cae en manos enemigas, negaremos su existencia. Si llega a manos amigas, continúen el trabajo: cada misión completada es una noche más bajo cielo humano. – General Luna.

