

Taller 4 Sistrans:

Tomás Sierra

Carlos Andres Peña Arias

Daniel Ortiz

Parte 1:

1. Las entidades se modelaron de dos maneras. Primero los bares tienen una lista de objetos embebidos las cuales representan las bebidas del bar. Esto puede servir para apoyar las consultas de las bebidas de un bar ya que rápidamente se puede consultar toda la información necesaria referente a las bebidas de un bar sobre un único documento. En cuanto a los bebedores esto es un documento aparte de los bares y cada bebedor tiene dos apuntadores referenciales a dos diferentes tipos de objetos: por un lado tiene una referencia a los bares frecuentados en forma de una lista de id 's. Por otro lado tiene una lista de preferencia de bebidas en forma de una lista con el nombre de las mismas.
2. Los tipos de consultas que pueden justificar las decisiones de modelado de parranderos no relacionales son: que se tenga a la mano la información de las bebidas asociadas a un bar, para formular un menú eficazmente. La consulta que más se ve priorizada frente a este diseño es la de las bebidas asociadas a un bar, la información asociada a un bar, sea su nombre, ciudad, presupuesto o cantidad de sedes. Finalmente información asociada a un bebedor. Se priorizan los requerimientos de consultas sobre los bares y bebedores. Se pierde prioridad en cuanto a las bebidas que les gustan a los bebedores, los bares frecuentados por los bebedores, las bebidas individuales y el tipo de todas las bebidas.

Ejemplos de Requerimiento de Consultas:

Consulta Priorizada: Obtener la lista de bebidas populares de un bar específico. Si el modelo embebe las bebidas dentro del documento del bar, esta consulta es directa y rápida.

Consulta No Priorizada: Encontrar todos los bebedores que visitan bares en dos ciudades específicas. Si los datos de bebedores y bares están separados, esta consulta podría requerir unión entre múltiples documentos y colecciones.

Parte 2:

1. Para modelar la nueva relación de dirección se tomó la decisión de que cada bar tiene una relación embebida con dirección. Entonces cada bar va a poder consultar dentro de su propio documento la dirección del bar sin necesidad de mezclar documentos. Esta

decisión se tomó ya que un bar va a tener en cualquier momento una sola dirección lo que permite que ese objeto no abrume el documento y apoye una consulta más fácil de direcciones.

2. Los esquemas finales están anexados con nombre de Bares.js y Bebedores.js:
3. Para generar la data para esta parte se usó Mockaroo.

Field Name	Type	Options
_id	Sequence	start at: 100 step: 1 repeat: 1 restart at: blank: 0 % Σ ×
nombre	Street Name	blank: 0 % Σ ×
ciudad	City	blank: 0 % Σ ×
presupuesto	JSON Array	min elements: 1 max elements: 1 blank: 0 % Σ ×
presupuesto.tipo_pr	Custom List	Alto, Medio, Bajo random blank: 0 % Σ ×
cant_sedes	Number	min: 1 max: 10 decimals: 0 blank: 0 % Σ ×
oferta_bebidas	JSON Array	min elements: 0 max elements: 5 blank: 0 % Σ ×
oferta_bebidas.nomb	Product (Grocery)	blank: 0 % Σ ×
oferta_bebidas.grad	Number	min: 0 max: 20 decimals: 0 blank: 0 % Σ ×
oferta_bebidas.tipo	Color	blank: 0 % Σ ×
oferta_bebidas.hora	Custom List	diurno, nocturno, todos random blank: 0 % Σ ×
oferta_bebidas.prec	Number	min: 500 max: 1000 decimals: 0 blank: 0 % Σ ×
direccion.ciudad	Template	{ciudad} ? blank: 0 % Σ ×
direccion.codigo_po	Number	min: 10000 max: 99999 decimals: 0 blank: 0 % Σ ×
direccion.direccion	Street Address	blank: 0 % Σ ×
direccion.telefono	Phone	format: ###-###-#### blank: 0 % Σ ×

El archivo se encuentra adjunto como VeinteBares.json

4. Para realizar la actualización de todos los documentos que no tienen dirección se usó la sentencia en UpdateBaresDireccion.js

Parte 3:

Para las consultas que puede encontrar anexadas en Consultas.js se va a usar una variable arbitraria X. Cada consulta va a estar marcada por su índice (1,2,3) para su ubicación en el archivo..