



Universidad Carlos III

Arquitectura de Datos

Curso 2024-25

Práctica 1

Diseño de una Base de Datos no relacional

Ingeniería Informática, Cuarto curso

Adrián Fernández Galán (NIA: 100472182, e-mail: 100472182@alumnos.uc3m.es)

César López Mantecón (NIA: 100472092, e-mail: 100472092@alumnos.uc3m.es)

Manuel Gómez-Plana Rodríguez (NIA: 100472092, e-mail: 100472092@alumnos.uc3m.es)

Prof . Lourdes Moreno López

Grupo: 81

Índice

1. Introducción	2
2. Diseño conceptual: diagrama de clases UML	2
3. Diseño de agregados	3
4. Validación del esquema	3

1. Introducción

2. Diseño conceptual: diagrama de clases UML

[Imagen del Modelo sin Agregados]

- Explicación del modelo básico sin los casos de uso y sin destacar las relaciones entre clases
 - Área (Centrándose en Tipo, lista de juegos, estado, N-juegos, clima)
 - Juego (Centrándose en Estado, Modelo, Historial de intervenciones, lista de incidencias)
- Análisis de los casos de uso y destacar decisiones en el diseño según estos casos
 - Caso de uso A
 - Este caso de uso está centrado en proporcionar un listado completo de los juegos instalados en las diferentes áreas
 - Para poder listar las áreas dado un distrito según el Req 1 se ha creado una entidad Distrito que facilite la búsqueda
 - Para satisfacer el Req 4 se ha añadido a la entidad Juego el atributo "patrón de desgaste"
 - Caso de uso B
 - Este Caso de uso está centrado en la capacidad del sistema de reportar incidencias
 - Se ha incorporado una entidad Incidencia con las características asociadas a los requisitos y otra entidad Usuarios
 - Para Req 2 se ha integrado en la entidad Incidencia el atributo lista de destinatarios, para conocer los usuarios que han realizado los reportes
 - Habría que explicar el por qué de todos los atributos de la entidad Incidencia, ahora mismo no caigo
 - Caso de Uso C
 - Este caso de uso está centrado en analizar las condiciones meteorológicas para realizar una planificación eficiente del mantenimiento
 - Para poder cubrir Req 1 se ha creado una entidad clima con los atributos "Temperatura", "Precipitación", "Fecha"
 - Para Req 2 y Req 3 no es necesario realizar ningún cambio dado que se conocen los aspectos meteorológicos de las áreas y por lo tanto de sus juegos
 - Req 4 queda fuera de nuestro alcance como diseñadores de bases de datos
 - Caso de Uso D
 - Este caso de uso busca mejorar la toma de decisiones sobre las áreas recreativas a través de la capacidad de generar informes y realizar encuestas de satisfacción por parte de los usuarios
 - Para cubrir Req 1 no es necesario introducir nuevas características ya que se contempló con anterioridad el uso incidencias
 - Se ha incorporado una entidad Encuesta para satisfacer Req 2, esta entidad tiene el usuario que lo ha realizado, la satisfacción que tiene con el juego y un comentario.
 - Para Req 3 proporcionamos la información de las áreas junto a los diferentes reportes
 - Caso de Uso E
 - Para Req 1 no es necesario incluir ninguna característica, ya que los cambios realizados para el caso de uso A ya contemplan este requisito
 - Para Req 2 se ha añadido el atributo "N-juegos" (esto no se podría calcular sin necesidad de tenerlo estático??)
 - Req 3 es viable hacerlo??
 - Req 4 queda fuera del nuestro alcance

3. Diseño de agregados

- Analizar las acciones (lectura, escritura y actualizaciones) de los casos de uso
- El caso de Uso A es completamente de lectura, ya que solo se quiere obtener información de las áreas y juegos

4. Validación del esquema

Para mantener la integridad de los datos, MongoDB ofrece el uso de estrategias como el uso de referencias en vez de embeber los datos. Sin embargo, para aquellos datos que hemos decidido embeber, debemos redactar una serie de reglas para mantener la consistencia de nuestro esquema. Estas reglas son:

- Nombres de *Distrito*: Cada distrito debe tener un nombre único para así evitar duplicados.
- Coordenadas de *Área*: Las coordenadas de las áreas serán una tupla que contenga en grados, minutos y segundos la latitud y longitud del área.
- Estado Operativo *Área*: El estado operativo de las áreas sólo podrán guardar los valores ".operativas", ".^{En} Mantenimiento" o "Fuera de servicio".
- Número de Juegos *Área*: El número de juegos instalados en un área debe ser un entero no negativo y siempre menor o igual al número máximo de juegos de un área.
- Número Máximo de Juegos *Área*: El número máximo de juegos de un área debe ser un entero no negativo.
- Temperatura *Clima*: La temperatura de Clima es un float y se medirá en grados Celsius.
- Precipitación *Clima*: La precipitación de Clima es un float y se medirá en litros por metro cuadrado.
- Estado Operativo *Juego*: El estado operativo de los juegos sólo podrá guardar dos valores "Disponible" o "No Disponible".
- QR *Juego*: El QR de Juego debe de ser una imagen.
- Accesibilidad *Juego*: La accesibilidad de un juego debe ser un texto que indique las personas con alguna deficiencia física, mental o sensorial que puedan acceder.