


## Semana 2 goals: (verde-completado, amarillo-a-medias, rojo-no-hecho) ¶

- CRUD de entidades principales en .NET (usuarios, reservas, espacios—y algunas tablas más en mi caso—). Esto lo hice en VS IDE al crear las clases y redactar el código relativo a los **modelos** (que se corresponden con las **tablas** de la **bd**), los **controller** (uno por cada modelo) y el **appdbcontext** (que declara los **dbsets** correspondientes a las tablas de la **bd**) y los **DTOs**. En las clases **controller** va el código de los **CRUD endpoints**. ¶ Implementación CRUD para gestión de reservas (crear, leer, actualizar, eliminar). ¶ Inicio de consumo básico de la base de datos en Docker. Si se refiere a que Angular frontend haga peticiones HTTP, todavía no lo tengo. Si se refiere a que por ejemplo Swagger haga operaciones CRUD con la bd de mi Docker sqlserver container, entonces lo pondría en amarillo. ¶
  - Documentación técnica de endpoints api: docu larga + docu resumen (recomendada, ya que se centro en lo actual y primordial). ¶ Documentación técnica parcial (API, esquema base de datos). Se trata de este pdf: ¶  
 EFCoreCode\_y\_PrimerosEndpoints.pdf (actualmente hay el LARGO y el RESUMEN) ¶
  - Pruebas unitarias en backend: estas pruebas hasta ahora las hice con Swagger y no todas. ¶ Primeras pruebas unitarias con xUnit (TDD) para servicios backend. También pendiente hacer pruebas empleando POSTMAN en lugar de Swagger.
  - Validaciones de negocio esenciales: me suena un poco abstracto, agradecería una mayor concreción, pero entiendo que estamos hablando de implementar cosas como el controlar que los datos introducidos por el usuario para su registro sean válidos (por ejemplo, que el dni cumple con su formato de 8 letras + 1 número). O que username y pw de login corresponden con los de un user de tabla de users. Tales validaciones yo no he hecho ninguna todavía. ¶
- Control de disponibilidad y validaciones en backend.
- Implementación de autenticación básica y control de permisos
  - Publicación del evento "ReservaCreada" en Kafka dummy/real. Publicación de eventos "ReservaCreada" hacia un topic Kafka simulado o real. Eventos iniciales Kafka. Instalé en mi proyecto backend la Confluent Kafka library y redacté la clase productora. ¶

Juan Carlos Martinez Perez jueves 15:19

👍 ❤️ 🤔 🤖 99 ...



### ### Semana 2: Backend y Base de Datos

- CRUD de entidades principales en .NET (usuarios, reservas, espacios)
- Implementación de autenticación básica y control de permisos
- Pruebas unitarias en backend y validaciones de negocio esenciales
- Publicación del evento "ReservaCreada" en Kafka dummy/real
- Documentación técnica de endpoints API