[Entity FrameWork Core](https://www.youtube.com/watch?v=rIqVDyHwmDE&t=1086s)

Passaggi:

1. Creo ApiWeb AspNet(credo sia piu semplice e abbia meno codice precompilato)
2. Una volta fatto mi assicuro di scaricare i nuggets dell'applicazione Mysql.EntityFrameworkCore e Entity….Core Tools
3. Creo la classe AppDbContext la quale eredita da DbContext e genero il costruttore che ha options (configurazioni delle connections string)
4. aggiungo nella classe Program il servizio
5. creo la cartella delle entità dove creo classi in base ai dati che voglio creare e aggiungo i dati public int/string/…. name {get; set;}
6. per farla diventare entità devo aggiungerla nel AppDbContext
7. una volta fatta aprire Console di gestione pacchetti che si trova in strumenti/gestione pacchetti e scrivere add-migration “name-della-migrazione”, Vedrai che si crea una cartella chiamata migrations con il codice che in teoria dovrebbe realizzare il database
8. scriviamo sul console update-database per realizzare i cambiamenti e dovrebbe crearsi il database su phpmyadmin e inoltre si crea una tabla history dove si vedono le migrazioni che si sono realizzate
9. in base ai nomi entityframework decide se i record stabilirli come primary key e anche i tipi o lunghezza(modifica per convenzione), Puoi aggiungere il tipo sopra le variabili delle entità [StringLength(maximumLength:10)](modifica per Annotazione)
10. è possibile anche usare api fluenti che sono di per se metodi, in appDbContext., inserire override onmodelcreating
11. si asegna i collegamenti tra gli id (uno con molti o molti con molti) hashset o di tipo classe…(da approfonfire)
12. possiamo anche creare una propria entità per la connessione tra due entità e dopo la entità potrà relazionarsi con la nuova tabella creata invece che con l'altra singola entità(minuto 56:00 del video)
13. Poi devo creare i Controllers (sono quelli che gestiscono le petizioni http  
    Creo il controller e eredito da ControlBase poi creo le petizioni
14. Al visualizzare si dovrebbero vedere tutti i dati delle entita per questo si usano i DTO(è una cartella dove inserisco solo le proprietà che voglio che si vedano nell'esecuzione dell’api) (si usa anche il pacchet auto mapper( minuto 1:11:00 del video)
15. Scarico il nugget automapper dependency injection e creo la classe e seleziono i valori che dorvrebbero essereci nella richiesta. Mi creo una cartella Utilita e creo la classe AutoMapperProfiles eredito da Profile ,ctor, e aggiungo il servizio al dbContext e dentro la classe inserisco classiDTO e le entita da dove prendere i dati