

ESTADO DEL PROYECTO Y ORDEN PARA AVANZAR

Normalización de Datos (4FN)		FRONTEND					BACKEND				DB
Entidad/Modelo	Foreign key	Funcio- nalidad(0)	Estilo	Valida- cion Inputs	Data Convert Object(1)	Servicio DATO(2)	Logica	Servicio API(3)	Patrón DTO	Integri- dad(4)	Persis- tencia
Person	1:1										
Interest	N:1 (1)										
Hardskill	N:1 (1)										
Softskill	N:1 (1)										
Project	N:1 (1)										
Phone	N:1 (1)										
SocialNetwork	N:1 (1)										
Organization											
Studie(*7)	N:1 (3)										
LaboralCareer(*7)	N:1 (3)										
Degree											
RolePosition											
DisplayData (*5)											
Card (*5)											
ProjectMedia(*6)	N:1 (1)										
		FRONTEND					BACKEND				DB
DESPLIEGUE											
ANULAR EndPoint usados en Test											
DOCUMENTACION (ACTUALIZADA)											DER
Interfase RESPONSIVA							Sin interfase				
REFACTORIZADO											
LOGGIN											
TOKEN											
GUARDS											

Referencias	
Verificado	Parcial
Completo	No iniciado
Modificado	

observaciones	
(0)	Alta, Baja, Modificacion y Visibilidad/Listado
(1)	Por cambios en el DER y nombres de atributos, debo crear clases para compatibilizar Modelos BackEnd con Clases FrontEnd. Creo que es mejor que multiples cambios en el codigo de las vistas.
(2)	Completo si apunta al Servicio API en localhost, parcial si apunta a una JSON.db
(3)	Funcionando en localhost, se obtienen los JSON y se actualiza en la Base de Datos
(4)	Completo si controla validaciones de atributos, parcial si controla solo Foreign key
(*5)	Elementos que permiten al usuario alterar el Layout del Portfolio (Visibilidad y estilo)
(*6)	Adicional para que destaque con capturas de pantallas o videos embebidos de los proyectos
(*7)	Complejidad para el formulario y manejo de fechas (widgets y validaciones)