Informe Técnico

1. Nombre del software

Sims 4 Db-Manager

2. Autores

- José Alejandro Solís Fernández 98060808501
- Bryan Machín García 98062508741
- Adrianna Álvarez Lorenzo 99072007576

3. Diccionario de Datos

Dado el conjunto de metadatos, el cual posee características lógicas de los datos que se utilizaron, se definió el siguiente estándar:

- Key(*Clave de la Tabla*)
- Name(*Nombre del Campo*)
- Data Type(*Tipo del Campo*)
- Allow Nulls(*Admite valor nulo*)
- Default(*Defecto*)

La especificación de cada tabla se muestra a continuación:

	Name	Data Type	Allow Nulls	Default
⚷	ActivityID	uniqueidentifier	☐	
	Name	nvarchar(127)	☐	
	Description	nvarchar(127)	☐	
			☐	

Keys (1)
 PK_Activities (Primary Key, Clustered: ActivityID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Figura 1: Diccionario de Datos Activities

	Name	Data Type	Allow Nulls	Default
⚷	ActivityID	uniqueidentifier	☐	
	SkillID	uniqueidentifier	☐	
			☐	

Keys (1)
 PK_ActivityImprovesSkill (Primary Key, Clustered: ActivityID)
Check Constraints (0)
Indexes (1)
 IX_ActivityImprovesSkill_SkillID (SkillID)
Foreign Keys (2)
 FK_ActivityImprovesSkill_Activities_ActivityID (ActivityID)
 FK_ActivityImprovesSkill_Skills_SkillID (SkillID)
Triggers (0)

Figura 2: Diccionario de Datos ActivityImprovesSkill

	Name	Data Type	Allow Nulls	Default
🔑	SkillID	uniqueidentifier	☐	
🔑	ActivityID	uniqueidentifier	☐	
	RequiredPoints	int	☐	
			☐	

Keys (1)
    PK_ActivityRequiresSkill (Primary Key, Clustered: SkillID, ActivityID)
Check Constraints (0)
Indexes (1)
    IX_ActivityRequiresSkill_ActivityID (ActivityID)
Foreign Keys (2)
    FK_ActivityRequiresSkill_Activities_ActivityID (ActivityID)
    FK_ActivityRequiresSkill_Skills_SkillID (SkillID)
Triggers (0)

Figura 3: Diccionario de Datos ActivityRequiresSkill

	Name	Data Type	Allow Nulls	Default
🔑	DomesticUnitID	uniqueidentifier	☐	
	Name	nvarchar(127)	☐	
	RoomNumber	int	☐	
	BathroomNumber	int	☐	
	Description	nvarchar(127)	☐	
			☐	

Keys (1)
    PK_DomesticUnits (Primary Key, Clustered: DomesticUnitID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Figura 4: Diccionario de Datos DomesticUnits

	Name	Data Type	Allow Nulls	Default
🔑	SimID	uniqueidentifier	☐	
	ProfessionID	uniqueidentifier	☐	
	Level	int	☐	
			☐	

Keys (1)
    PK_Exercises (Primary Key, Clustered: SimID)
Check Constraints (0)
Indexes (1)
    IX_Exercises_ProfessionID (ProfessionID)
Foreign Keys (2)
    FK_Exercises_Professions_ProfessionID (ProfessionID)
    FK_Exercises_Sims_SimID (SimID)
Triggers (0)

Figura 5: Diccionario de Datos Exercises

	Name	Data Type	Allow Nulls	Default
🔑	QuestID	uniqueidentifier	☐	
🔑	SimID	uniqueidentifier	☐	
🔑	WorldID	uniqueidentifier	☐	
🔑	Date	datetime2(7)	☐	
	Success	bit	☐	(CONVERT([bit],(0)))
			☐	

Keys (1)
    PK_Involvements (Primary Key, Clustered: SimID, Date, QuestID, WorldID)
Check Constraints (0)
Indexes (2)
    IX_Involvements_QuestID (QuestID)
    IX_Involvements_SimID_WorldID_Date (SimID, WorldID, Date)
Foreign Keys (2)
    FK_Involvements_Quests_QuestID (QuestID)
    FK_Involvements_Travels_SimID_WorldID_Date (SimID, WorldID, Date)
Triggers (0)

Figura 6: Diccionario de Datos Involvements

Name	Data Type	Allow Nulls	Default
DomesticUnitID	uniqueidentifier	☐	
NeighborhoodID	uniqueidentifier	☐	
		☐	

Keys (1)
    PK_NeighborhoodDomesticUnits   (Primary Key, Clustered: DomesticUnitID)
Check Constraints (0)
Indexes (1)
    IX_NeighborhoodDomesticUnits_NeighborhoodID   (NeighborhoodID)
Foreign Keys (2)
    FK_NeighborhoodDomesticUnits_DomesticUnits_DomesticUnitID (DomesticUnitID)
    FK_NeighborhoodDomesticUnits_Neighborhoods_NeighborhoodID (NeighborhoodID)
Triggers (0)

Figura 7: Diccionario de Datos NeighborhoodDomesticUnits

Name	Data Type	Allow Nulls	Default
PlaceID	uniqueidentifier	☐	
NeighborhoodID	uniqueidentifier	☐	
		☐	

Keys (1)
    PK_NeighborhoodPlaces   (Primary Key, Clustered: PlaceID)
Check Constraints (0)
Indexes (1)
    IX_NeighborhoodPlaces_NeighborhoodID   (NeighborhoodID)
Foreign Keys (2)
    FK_NeighborhoodPlaces_Neighborhoods_NeighborhoodID   (NeighborhoodID)
    FK_NeighborhoodPlaces_Places_PlaceID   (PlaceID)
Triggers (0)

Figura 8: Diccionario de Datos NeighborhoodPlaces

Name	Data Type	Allow Nulls	Default
NeighborhoodID	uniqueidentifier	☐	
Name	nvarchar(127)	☐	
Description	nvarchar(127)	☐	
		☐	

Keys (1)
    PK_Neighborhoods   (Primary Key, Clustered: NeighborhoodID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Figura 9: Diccionario de Datos Neighborhoods

Name	Data Type	Allow Nulls	Default
NeighborhoodID	uniqueidentifier	☐	
SkillID	uniqueidentifier	☐	
		☐	

Keys (1)
    PK_NeighborhoodUpgradesSkill   (Primary Key, Clustered: NeighborhoodID)
Check Constraints (0)
Indexes (1)
    IX_NeighborhoodUpgradesSkill_SkillID   (SkillID)
Foreign Keys (2)
    FK_NeighborhoodUpgradesSkill_Neighborhoods_NeighborhoodID   (NeighborhoodID)
    FK_NeighborhoodUpgradesSkill_Skills_SkillID   (SkillID)
Triggers (0)

Figura 10: Diccionario de Datos NeighborhoodUpgradesSkill

Name	Data Type	Allow Nulls	Default
SimID	uniqueidentifier	☐	
ActivityID	uniqueidentifier	☐	
LastPerform	datetime2(7)	☐	
		☐	

Keys (1)
  PK_Performances (Primary Key, Clustered: SimID, ActivityID)
Check Constraints (0)
Indexes (1)
  IX_Performances_ActivityID (ActivityID)
Foreign Keys (2)
  FK_Performances_Activities_ActivityID (ActivityID)
  FK_Performances_Sims_SimID (SimID)
Triggers (0)

Figura 11: Diccionario de Datos Performances

Name	Data Type	Allow Nulls	Default
PetID	uniqueidentifier	☐	
DomesticUnitID	uniqueidentifier	☐	
		☐	

Keys (1)
  PK_PetLives (Primary Key, Clustered: PetID)
Check Constraints (0)
Indexes (1)
  IX_PetLives_DomesticUnitID (DomesticUnitID)
Foreign Keys (2)
  FK_PetLives_DomesticUnits_DomesticUnitID (DomesticUnitID)
  FK_PetLives_Pets_PetID (PetID)
Triggers (0)

Figura 12: Diccionario de Datos PetLives

Name	Data Type	Allow Nulls	Default
PetID	uniqueidentifier	☐	
Name	nvarchar(127)	☐	
TypeName	nvarchar(MAX)	☑	
TypeID	uniqueidentifier	☑	
		☐	

Keys (1)
  PK_Pets (Primary Key, Clustered: PetID)
Check Constraints (0)
Indexes (1)
  IX_Pets_TypeID (TypeID)
Foreign Keys (1)
  FK_Pets_PetTypes_TypeID (TypeID)
Triggers (0)

Figura 13: Pets

Name	Data Type	Allow Nulls	Default
TypeID	uniqueidentifier	☐	
Name	nvarchar(127)	☐	
Description	nvarchar(127)	☐	
		☐	

Keys (1)
  PK_PetTypes (Primary Key, Clustered: TypeID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Figura 14: Diccionario de Datos PetTypes

Name	Data Type	Allow Nulls	Default
PlaceID	uniqueidentifier	☐	
Name	nvarchar(127)	☐	
Cost	float	☐	
Description	nvarchar(127)	☐	
		☐	

Keys (1)
  PK_Places   (Primary Key, Clustered: PlaceID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Figura 15: Diccionario de Datos Places

Name	Data Type	Allow Nulls	Default
ProfessionID	uniqueidentifier	☐	
Name	nvarchar(127)	☐	
BasicSalary	float	☐	
		☐	

Keys (1)
  PK_Professions   (Primary Key, Clustered: ProfessionID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Figura 16: Professions

Name	Data Type	Allow Nulls	Default
ProfessionID	uniqueidentifier	☐	
SkillID	uniqueidentifier	☐	
		☐	

Keys (1)
  PK_ProfessionUpgradesSkill   (Primary Key, Clustered: ProfessionID)
Check Constraints (0)
Indexes (1)
  IX_ProfessionUpgradesSkill_SkillID   (SkillID)
Foreign Keys (2)
  FK_ProfessionUpgradesSkill_Professions_ProfessionID   (ProfessionID)
  FK_ProfessionUpgradesSkill_Skills_SkillID   (SkillID)
Triggers (0)

Figura 17: Diccionario de Datos ProfessionUpgradesSkill

Name	Data Type	Allow Nulls	Default
SkillID	uniqueidentifier	☐	
QuestID	uniqueidentifier	☐	
RequiredPoints	int	☐	
		☐	

Keys (1)
  PK_QuestRequiresSkill   (Primary Key, Clustered: SkillID, QuestID)
Check Constraints (0)
Indexes (1)
  IX_QuestRequiresSkill_QuestID   (QuestID)
Foreign Keys (2)
  FK_QuestRequiresSkill_Quests_QuestID   (QuestID)
  FK_QuestRequiresSkill_Skills_SkillID   (SkillID)
Triggers (0)

Figura 18: Diccionario de Datos QuestRequiresSkill

	Name	Data Type	Allow Nulls	Default
🔑	QuestID	uniqueidentifier	☐	
	Name	nvarchar(127)	☐	
	Cost	float	☐	
	Reward	float	☐	
	Description	nvarchar(127)	☐	(N'')
			☐	

Keys (1)
PK_Quests   (Primary Key, Clustered: QuestID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Figura 19: Diccionario de Datos Quests

	Name	Data Type	Allow Nulls	Default
🔑	QuestID	uniqueidentifier	☐	
	WorldID	uniqueidentifier	☐	
			☐	

Keys (1)
PK_QuestWorld   (Primary Key, Clustered: QuestID)
Check Constraints (0)
Indexes (1)
IX_QuestWorld_WorldID   (WorldID)
Foreign Keys (2)
FK_QuestWorld_Quests_QuestID   (QuestID)
FK_QuestWorld_Worlds_WorldID   (WorldID)
Triggers (0)

Figura 20: Diccionario de Datos QuestWorld

	Name	Data Type	Allow Nulls	Default
🔑	SimID	uniqueidentifier	☐	
	DomesticUnitID	uniqueidentifier	☐	
			☐	

Keys (1)
PK_SimLives   (Primary Key, Clustered: SimID)
Check Constraints (0)
Indexes (1)
IX_SimLives_DomesticUnitID   (DomesticUnitID)
Foreign Keys (2)
FK_SimLives_DomesticUnits_DomesticUnitID   (DomesticUnitID)
FK_SimLives_Sims_SimID   (SimID)
Triggers (0)

Figura 21: Diccionario de Datos SimLives

	Name	Data Type	Allow Nulls	Default
🔑	SimID	uniqueidentifier	☐	
	Name	nvarchar(127)	☐	
	LastName	nvarchar(127)	☐	
	Money	float	☐	
	Gender	nvarchar(MAX)	☑	
	LifeStage	nvarchar(MAX)	☑	
			☐	

Keys (1)
PK_Sims   (Primary Key, Clustered: SimID)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Figura 22: Diccionario de Datos Sims

Figura 23: Diccionario de Datos SimSkills

Figura 24: Diccionario de Datos Skills

Figura 25: Diccionario de Datos Travels

Figura 26: Diccionario de Datos Worlds

	Name	Data Type	Allow Nulls	Default
🔑	MigrationId	nvarchar(150)	☐	
	ProductVersion	nvarchar(32)	☐	
			☐	

Keys (1)
PK__EFMigrationsHistory   (Primary Key, Clustered: MigrationId)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)

Figura 27: Diccionario de Datos __ EFMigrationsHistory

A continuación se muestran las dependencias entre dichos Diccionarios de Datos:
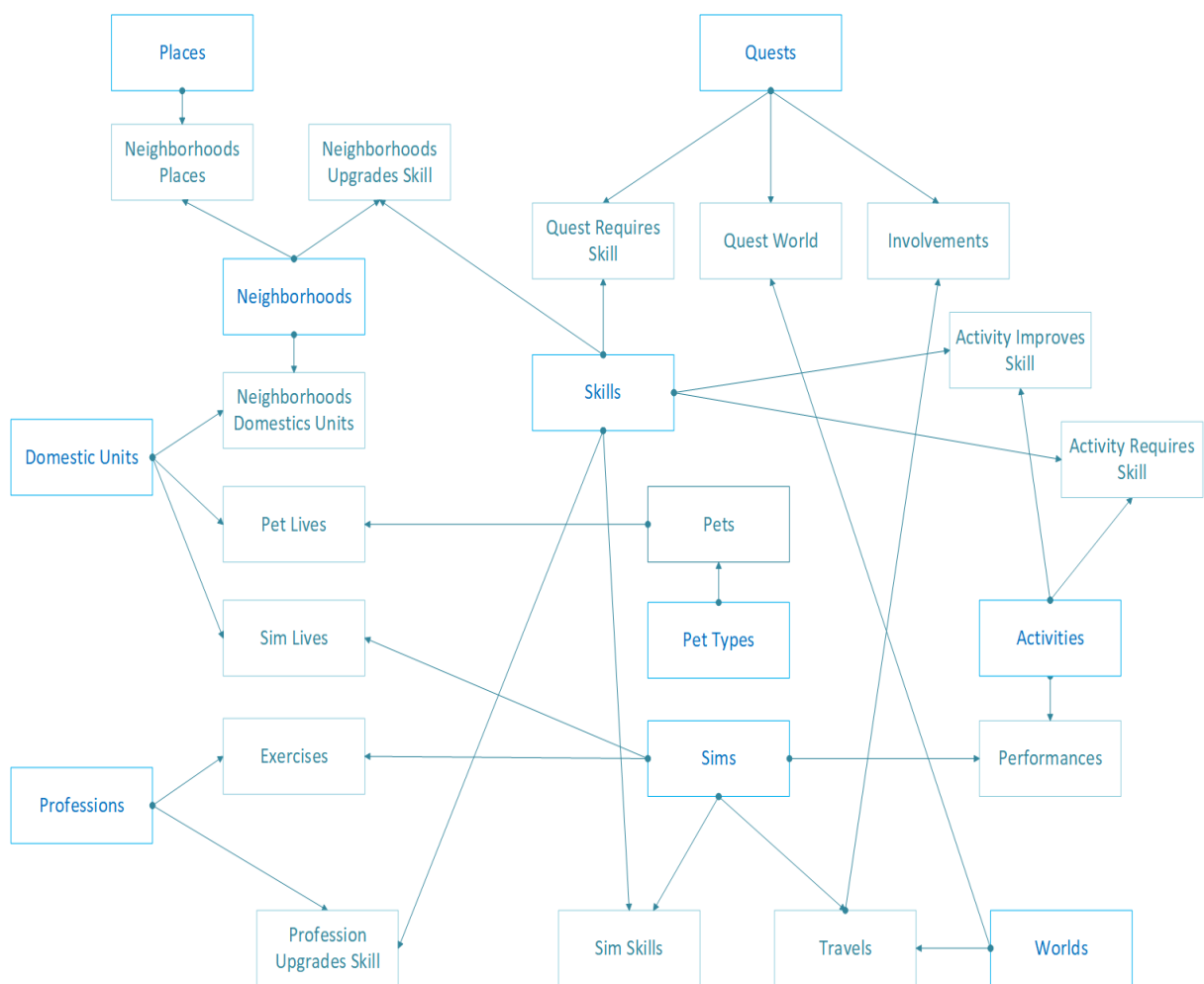
Figura 28: Dependencias entre los Diccionarios de Datos

4. Esquema con el diseño de la aplicación

La aplicación implementa un patrón de arquitectura de software Modelo-Vista-Controlador, utilizando EntityFramework para la implementación de la capa de acceso a datos, teniendo así entidades de modelo que pueden ser utilizadas por todas las capas de la aplicación, y además se emplea el patrón Repositorio como capa de abstracción entre el acceso a los datos y la capa de lógica de negocio de la aplicación.
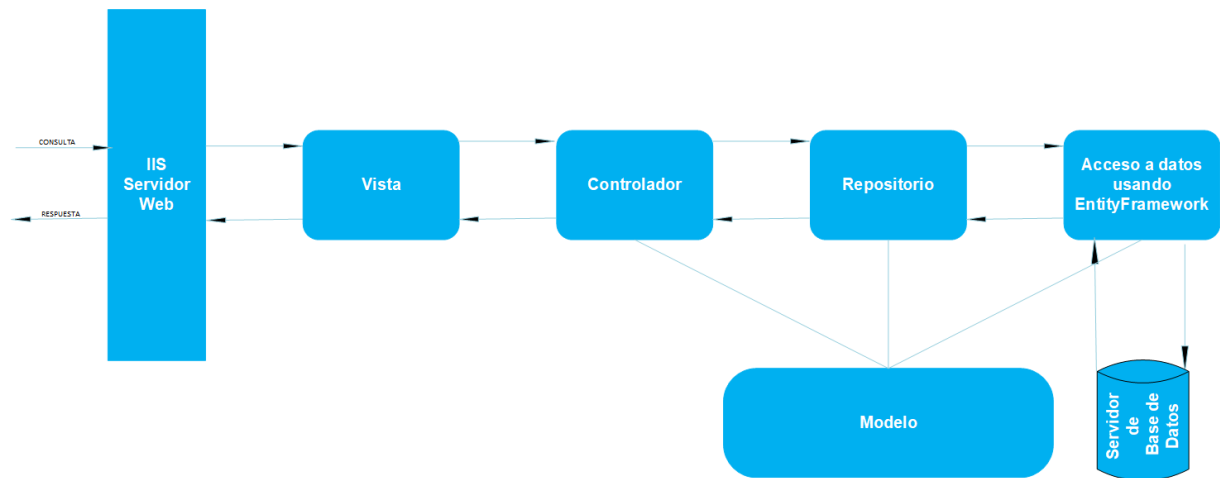
Figura 29: Diseño de la Aplicación

El Modelo incluye la lógica para manipular, gestionar y actualizar los datos de la base de datos, cuyo Sistema de Gestión empleado fue SQL Server Express LocalDB. Para ello se definió la interface IRepository que se encuentra dentro del *namespace Sims.Models.Data*, en la que las clases que dependan de ella pueden obtener estos objetos sin necesidad de conocer cómo se almacenan o cómo la clase de implementación los devolverá.

Los Controladores se encargan de gestionar las instrucciones que reciben y procesarlas, conteniendo así la lógica de negocio de la aplicación. Este utiliza una *inyección de dependencia(dependency injection)*, pues su enfoque permite al constructor de cada clase controller definida, acceder al repositorio de la aplicación a través de la interfaz IRepository sin necesidad de conocer cuál clase de implementación se ha configurado.

Como Entity FrameWork Core puede generar el esquema para la base de datos utilizando las clases del modelo, se creó una migración de la base de datos. Luego, si se necesita modificar las clases del modelo, simplemente se puede crear una nueva migración que contenga los comandos SQL necesarios para reflejar dichos cambios.

5. Esquema con las clases definidas

Figura 30: Controllers

Figura 31: Migrations

Figura 32: Models

ApplicationDbContext
Class
→ DbContext

▲ Properties
- Activities
- ActivityImprovesSkill
- ActivityRequiresSkill
- DomesticUnits
- Exercises
- Involvements
- NeighborhoodDomesticUnits
- NeighborhoodPlaces
- Neighborhoods
- NeighborhoodUpgradesSkill
- Performances
- PetLives
- Pets
- PetTypes
- Places
- Professions
- ProfessionUpgradesSkill
- QuestRequiresSkill
- Quests
- QuestWorld
- SimLives
- Sims
- SimSkills
- Skills
- Travels
- Worlds

▲ Methods
- ApplicationDbContext
- OnModelCreating

DbContext
Class

IInfrastructure<IServiceProvider>
IDbContextDependencies
IDbSetCache
IDbContextPoolable
IResettableService

▲ Properties
- ChangeTracker
- ContextId
- Database
- Model

▲ Methods
- Add<TEntity> (+ 1 overload)
- AddAsync<TEntity> (+ 1 overload)
- AddRange (+ 1 overload)
- Attach<TEntity> (+ 1 overload)
- AttachRange (+ 1 overload)
- DbContext (+ 1 overload)
- Dispose
- DisposeAsync
- Entry<TEntity> (+ 1 overload)
- Equals
- Find (+ 1 overload)
- FindAsync (+ 3 overloads)
- FromExpression<TResult>
- GetHashCode
- OnConfiguring
- OnModelCreating
- Remove<TEntity> (+ 1 overload)
- RemoveRange (+ 1 overload)
- SaveChanges (+ 1 overload)
- SaveChangesAsync (+ 1 overload)
- Set<TEntity> (+ 1 overload)
- ToString
- Update<TEntity> (+ 1 overload)
- UpdateRange (+ 1 overload)

▲ Events
- SaveChangesFailed
- SavedChanges
- SavingChanges

EFRepository
Class

IRepository

▲ Fields
- context

▲ Properties
- Activities
- ActivityImprovesSkillTable
- ActivityRequiresSkillsTable
- DomesticUnits
- Exercises
- Involvements
- NeighborhoodDomesticUnitsTable
- NeighborhoodPlacesTable
- Neighborhoods
- NeighborhoodUpgradesSkillsTable
- Performances
- PetLivesTable
- Pets
- PetTypes
- Places
- Professions
- ProfessionUpgradesSkillsTable
- QuestRequiresSkillTable
- Quests
- QuestWorldTable
- SimLivesTable
- Sims
- SimSkillsTable
- Skills
- Travels
- Worlds

▲ Methods
- DeleteActivity
- DeleteActivityImprovesSkill
- DeleteActivityRequiresSkill
- DeleteDomesticUnit
- DeleteExercise
- DeleteInvolve
- DeleteNeighborhood
- DeleteNeighborhoodDomesticUnit
- DeleteNeighborhoodPlace
- DeleteNeighborhoodUpgradesSkill
- DeletePerform
- DeletePet
- DeletePetLives
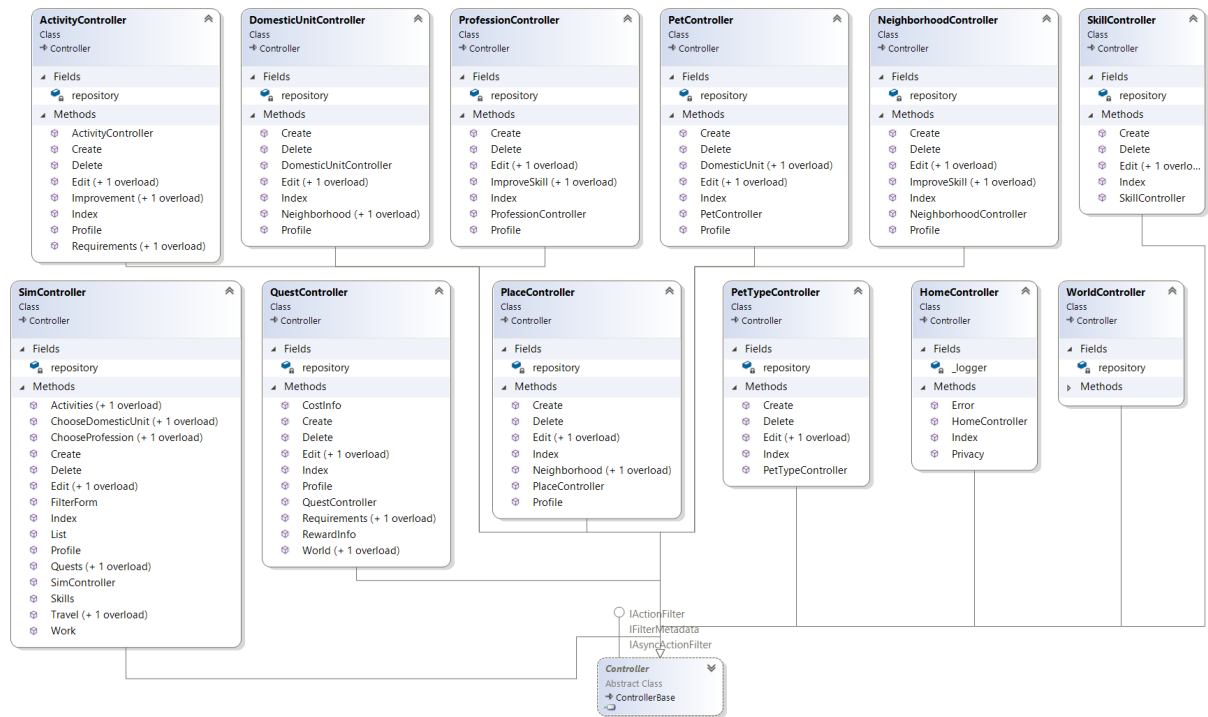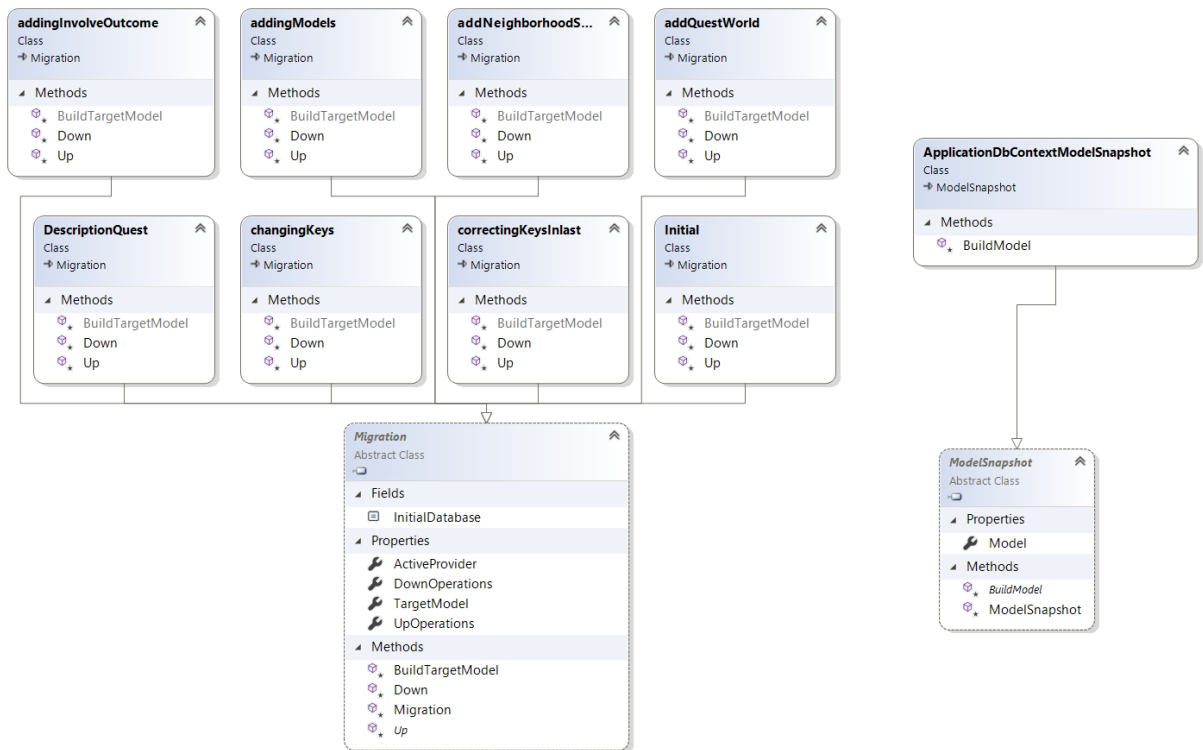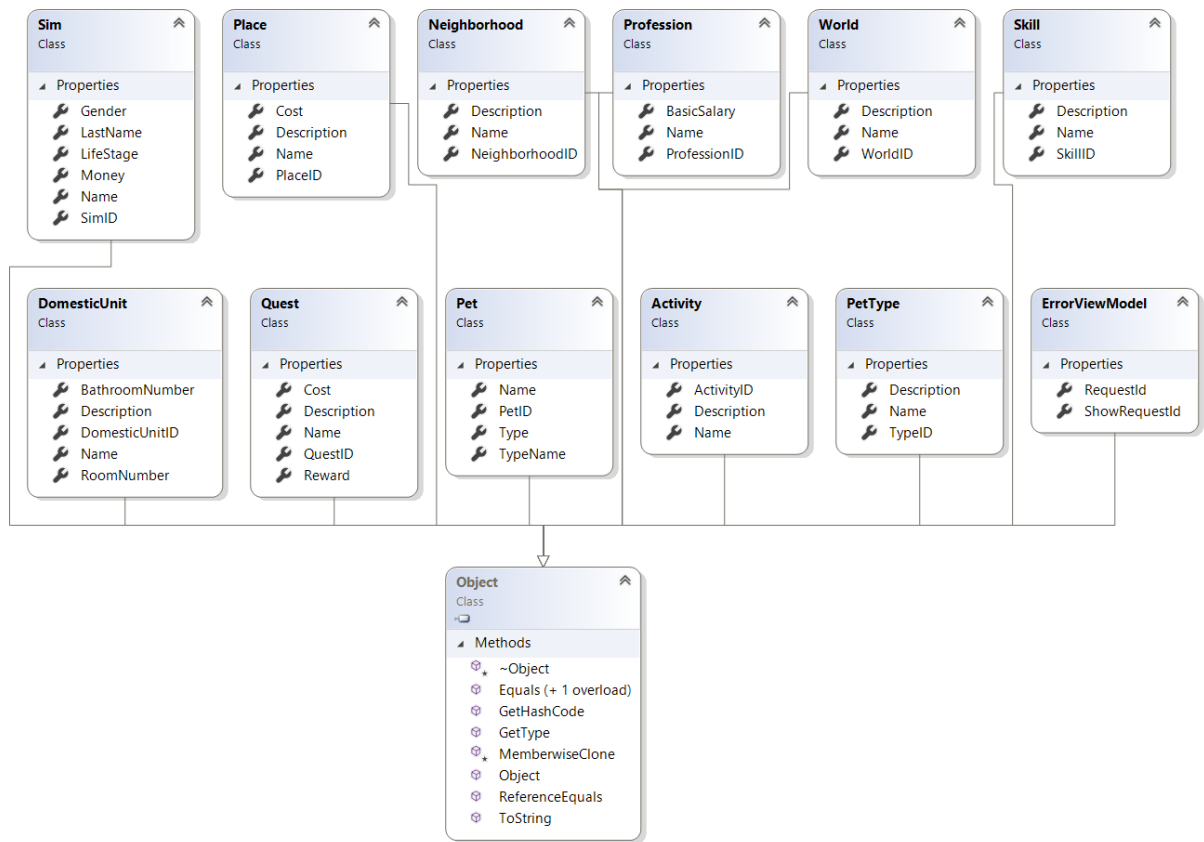- DeletePetType
- DeletePlace
- DeleteProfession
- DeleteProfessionUpgradesSkill
- DeleteQuest
- DeleteQuestRequiresSkill
- DeleteQuestWorld
- DeleteSim
- DeleteSimLives
- DeleteSimSkills
- DeleteSkill
- DeleteTravel
- DeleteWorld
- EFRepository
- SaveActivity
- SaveActivityImprovesSkill
- SaveActivityRequiresSkill
- SaveDomesticUnit
- SaveExercise
- SaveInvolve
- SaveNeighborhood
- SaveNeighborhoodDomesticUnit
- SaveNeighborhoodPlace
- SaveNeighborhoodUpgradesSkill
- SavePerform
- SavePet
- SavePetLives
- SavePetType
- SavePlace
- SaveProfession
- SaveProfessionUpgradesSkill
- SaveQuest
- SaveQuestRequiresSkill
- SaveQuestWorld
- SaveSim
- SaveSimLives
- SaveSimSkills
- SaveSkill
- SaveTravel
- SaveWorld

Object
Class

▲ Methods
- ~Object
- Equals (+ 1 overload)
- GetHashCode
- GetType
- MemberwiseClone
- Object
- ReferenceEquals
- ToString

IRepository
Interface

▲ Properties
- Activities
- ActivityImprovesSkillTable
- ActivityRequiresSkillsTable
- DomesticUnits
- Exercises
- Involvements
- NeighborhoodDomesticUnitsTable
- NeighborhoodPlacesTable
- Neighborhoods
- NeighborhoodUpgradesSkillsTable
- Performances
- PetLivesTable
- Pets
- PetTypes
- Places
- Professions
- ProfessionUpgradesSkillsTable
- QuestRequiresSkillTable
- Quests
- QuestWorldTable
- SimLivesTable
- Sims
- SimSkillsTable
- Skills
- Travels
- Worlds

▲ Methods
- DeleteActivity
- DeleteActivityImprovesSkill
- DeleteActivityRequiresSkill
- DeleteDomesticUnit
- DeleteExercise
- DeleteInvolve
- DeleteNeighborhood
- DeleteNeighborhoodDomesticUnit
- DeleteNeighborhoodPlace
- DeleteNeighborhoodUpgradesSkill
- DeletePerform
- DeletePet
- DeletePetLives
- DeletePetType
- DeletePlace
- DeleteProfession
- DeleteProfessionUpgradesSkill
- DeleteQuest
- DeleteQuestRequiresSkill
- DeleteQuestWorld
- DeleteSim
- DeleteSimLives
- DeleteSimSkills
- DeleteSkill
- DeleteTravel
- DeleteWorld
- SaveActivity
- SaveActivityImprovesSkill
- SaveActivityRequiresSkill
- SaveDomesticUnit
- SaveExercise
- SaveInvolve
- SaveNeighborhood
- SaveNeighborhoodDomesticUnit
- SaveNeighborhoodPlace
- SaveNeighborhoodUpgradesSkill
- SavePerform
- SavePet
- SavePetLives
- SavePetType
- SavePlace
- SaveProfession
- SaveProfessionUpgradesSkill
- SaveQuest
- SaveQuestRequiresSkill
- SaveQuestWorld
- SaveSim
- SaveSimLives
- SaveSimSkills
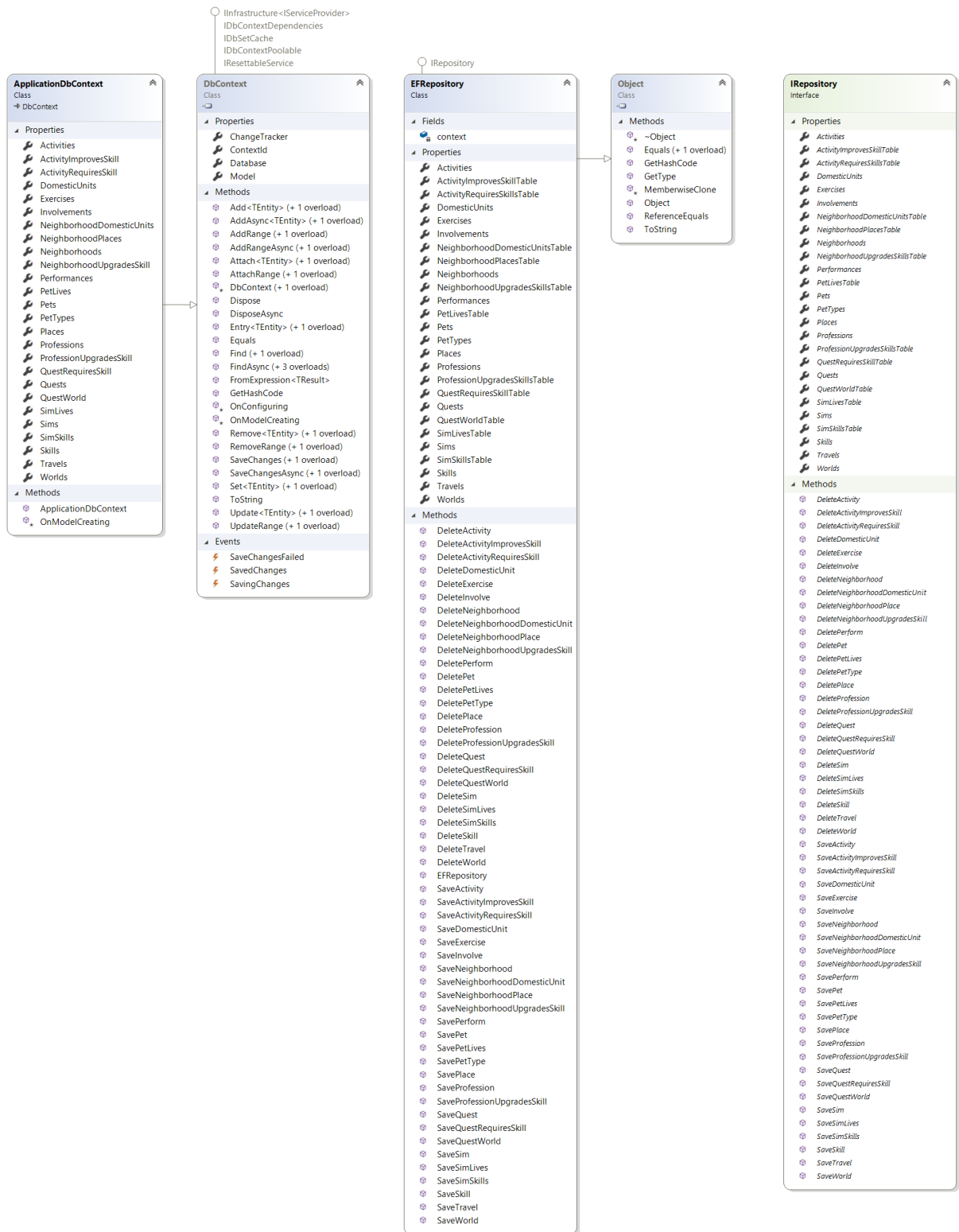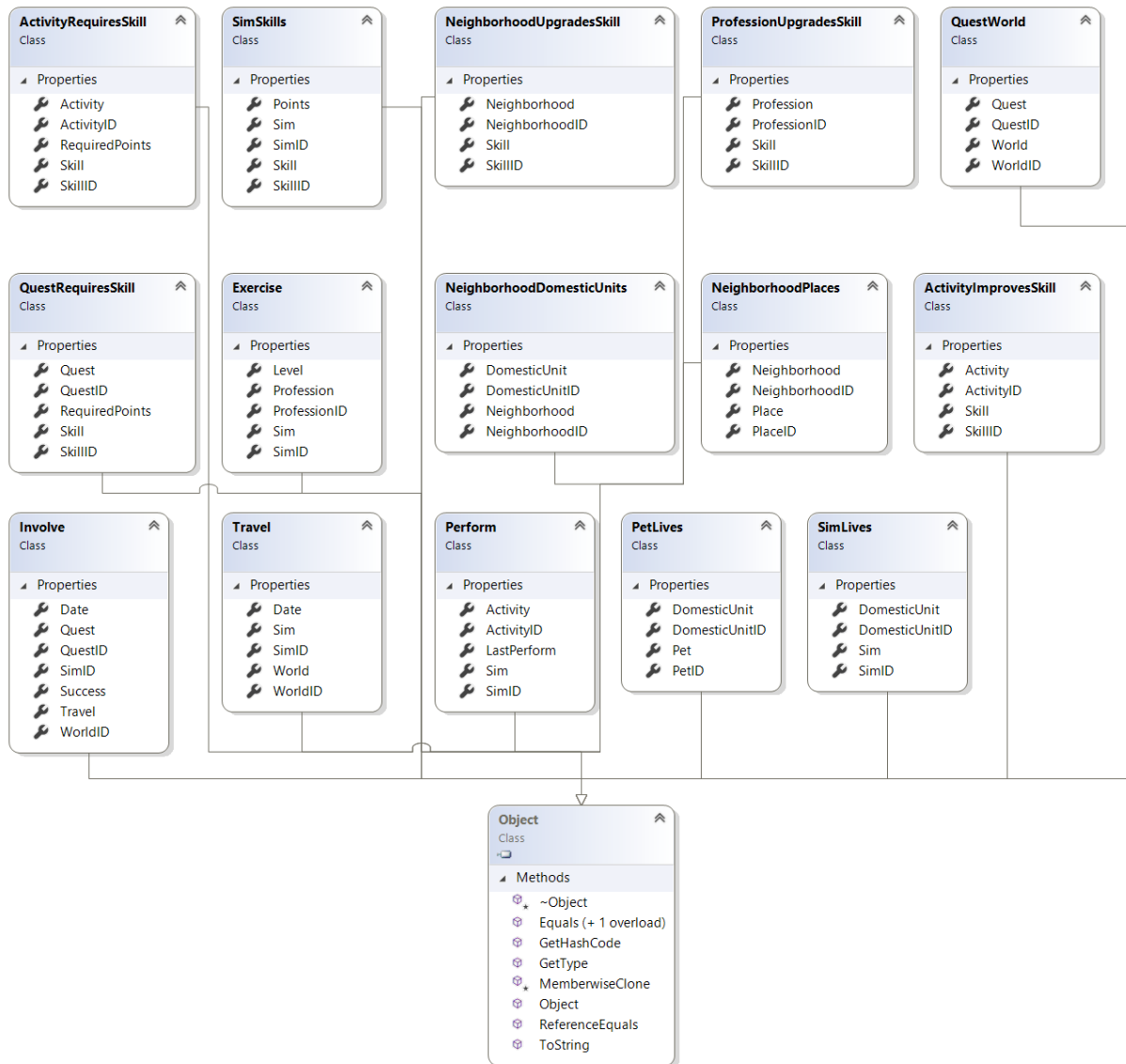- SaveSkill
- SaveTravel
- SaveWorld

Figura 33: Models Data

Figura 34: Models Relations

Figura 35: View Models