Universidad ORT Uruguay

Facultad de Ingeniería

Diseño de Aplicaciones 1

Entrega del Obligatorio 2

Joel Piña 228901

Ignacio Galisteo 237092

Grupo M4A

Docente: Syfer – Souto

Entregado como requisito de la materia Diseño de Aplicaciones 1

21 de noviembre de 2019

Tabla de contenido

Abstract

El siguiente documento es entregado como requerimiento para la materia diseño de aplicaciones 1. Se presenta una solución para la propuesta planteada para el segundo obligatorio de la misma materia.

Descripción general del trabajo y del sistema

El trabajo se enmarca en la implementación de un sistema de reserva de estacionamiento mediante el uso de mensajes de texto (SMS) mediante una interfaz gráfica.
En el sistema se pueden registrar cuentas que son identificadas mediante el número celular, estas pueden ser para distintas nacionalidades. Para la presente entrega se implementaron para Uruguay y Argentina. Las cuentas de Uruguay cuentan con el siguiente formato 09X XXX XXX o 9X XXX XXX, y para las cuentas de Argentina XXXXXX o XXXXXXX o XXXXXXXX.
En las siguientes cuentas se puede ingresar saldo mediante la UI, ingresando el número de móvil asociado a la cuenta, si el saldo es mayor a 0 se aumenta el saldo en la cuenta.
Los usuarios pueden reservar estacionamiento mediante un SMS ingresando la matrícula de su auto, la cantidad de minutos que desean reservar este debe ser un múltiplo de 30 y la hora que se desea que comience la reserva (este último es opcional).
También se implementan funcionalidades para que los inspectores de tránsito puedan consultar las reservas.

1. Se ingresa matrícula y hora para verificar si tiene reserva
2. Se ingresa un lapso de tiempo y se muestran todas las matrículas que tienen reservas.
3. Se ingresa un lapso de tiempo, una matrícula y el sistema filtra las reservas para esa matrícula en ese lapso de tiempo si las hay.
4. También se puede filtrar por país.

Descripción y Justificación de diseño

Creemos que las dos principales decisiones de diseño fueron:

1. Una arquitectura con múltiples capas. Tomando (sin llegar a su nivel de sofisticación) algunos aspectos de la “clean architecture” donde los distintos niveles se van referenciando hacia el interior. Y donde el centro contiene las “Entities” y las diversas interfaces.

2. La utilización de un “Repository Pattern” y un “Unit of Work Pattern”. Tenemos una interfaz “IRepository” con su correspondiente implementación, donde se encuentran algunas de las operaciones más comunes de acceso a la base de datos. Y otras interfaces e implementaciones que extienden la anterior y contienen operaciones más específicas relativas a sus entidades. También existe una interfaz e implementación de la “Unit of Work”. Que contiene los distintos repositorios y el método “Save” que encapsula el guardado de operaciones en el contexto.
   La utilización de estos patrones nos brinda como beneficio un mayor nivel de abstracción, y permitiría, por ejemplo, cambiar el mecanismo de persistencia sin tener que introducir mayores cambios de diseño.

3. En lo referente a las validaciones (en particular sms y números telefónicos) el diseño siguiendo un “Factory Method Pattern”.
   Tenemos interfaces para los distintos tipos de validadores. A modo de ejemplo: la interfaz “ICellPhoneValidator. Esta obliga a las clases que la implementen a tener una referencia a su país. Luego, existe una clase que retorna un objeto que es de tipo “ICellPhoneValidator” basándose en el país actual que se le pasa como parámetro. En este caso concreto, dicha clase

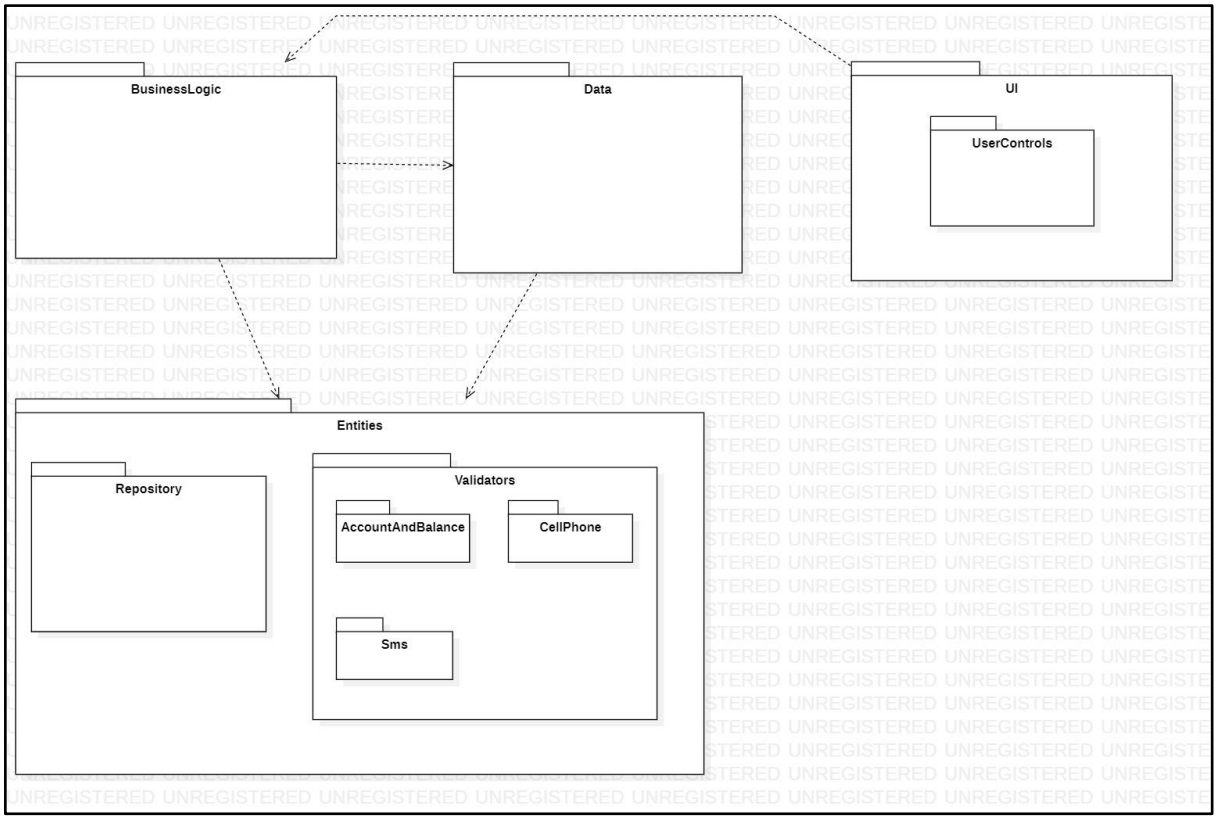
es “Account”, porque entendimos que era la más adecuada para tener tal responsabilidad. Así, si le pasamos un país con nombre “Argentina”, buscará en los validadores guardados en memoria el que contenga una referencia a dicho país y devolverá un “ArgCellPhoneValidator”.

4. También se recurrió a “dependency injection” en algún punto de la solución, para afrontar problemas concretos. Por ejemplo, para efectuar inyectamos al Parking Systema una “IUnitOfWork” a través de su constructor. A su vez, hay una helper class que, según el ambiente que se le pasa, devuelve una Unit of Work configurada de manera distinta, según ese ambiente sea de test o de desarrollo. Otra opción hubiera sido implementar una clase “UnitOfWorkTest” para inyectar las dependencias del ambiente de testeo.

   También diseñamos nuestro “ParkingContext”, de manera tal que recibe en su constructor un “IDataBaseInitializer<ParkingContext>” lo que nos permite llevar a cabo la solución del párrafo anterior.

Diagrama de paquetes

Se presenta el diagrama de paquetes realizado para el actual estado general organizacional del sistema.



Se adjunta el diagrama en formato PDF en el repositorio y en el archivo entregado para una mejor visualización.

Diagramas de clases

Se realizaron los diagramas para los paquetes Business Logic, Data y Entities. A continuación, se pueden apreciar al detalle. Los siguientes diagramas se incluyen tanto en el repositorio como en el archivo entregado para una mejor visualización.

Diagrama de clases de BusinessLogic

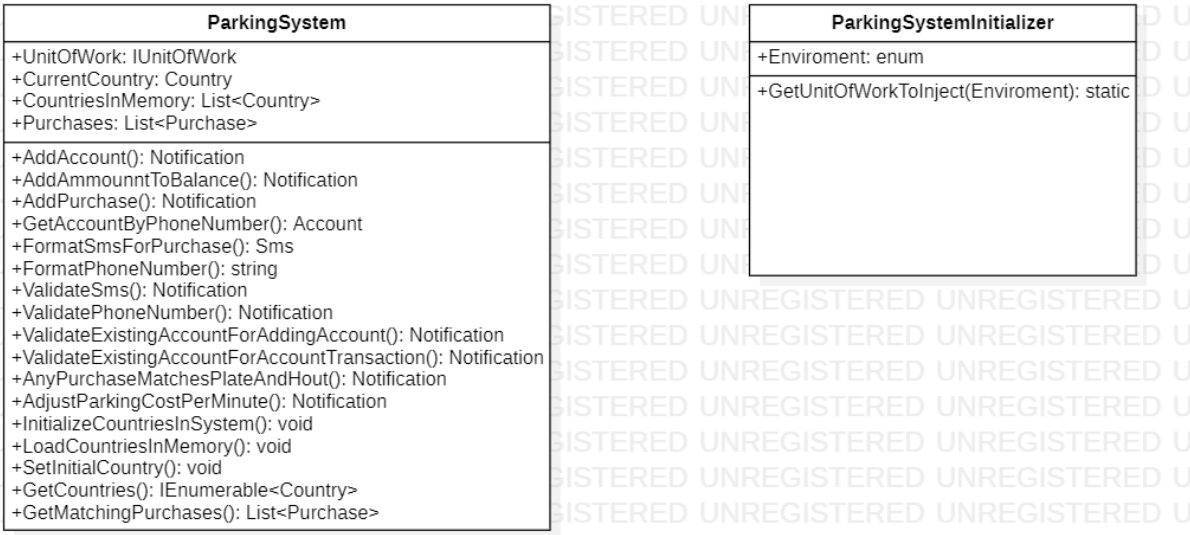


Diagrama de clases de Data

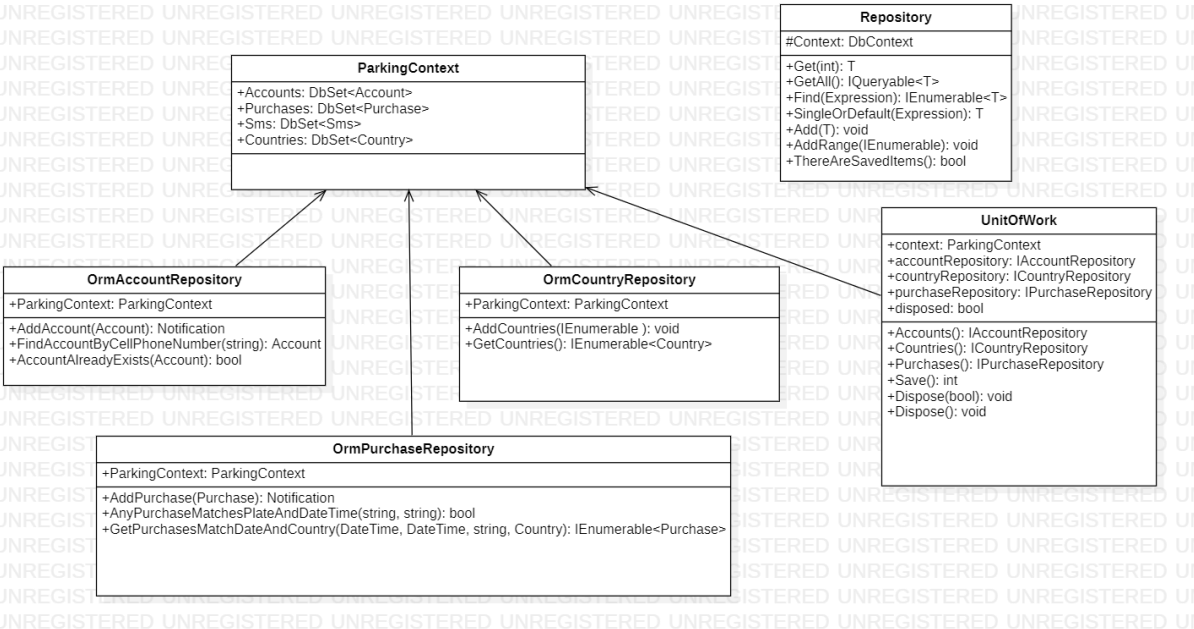


Diagrama de clases Entities

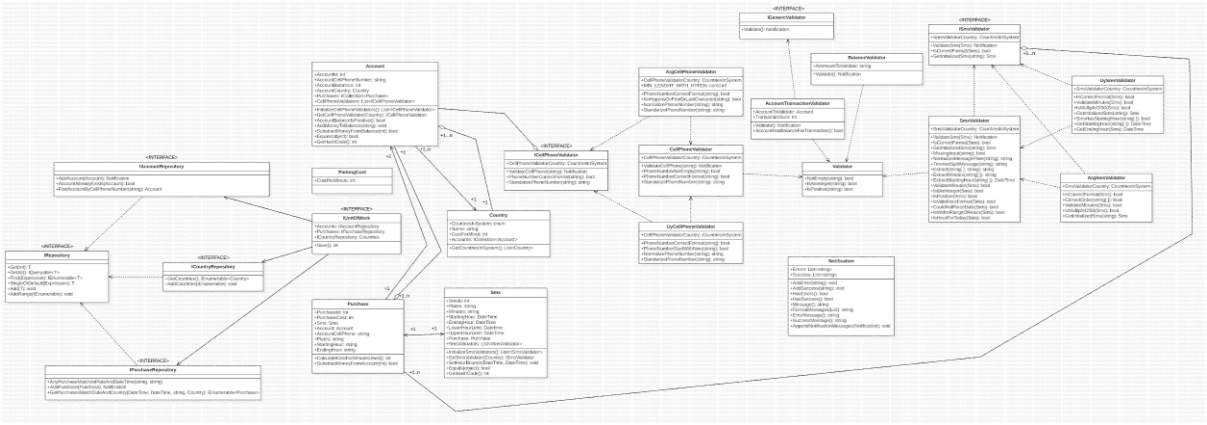
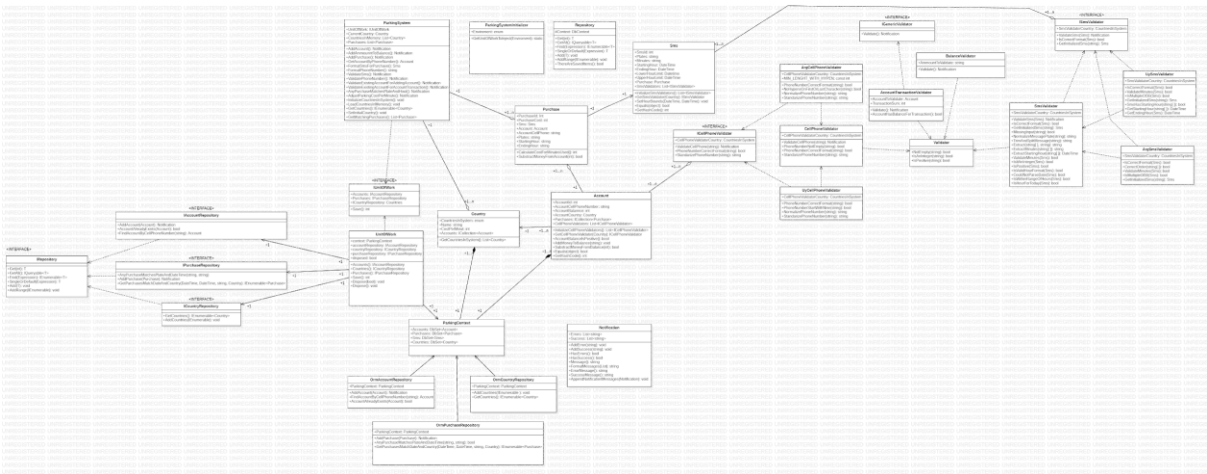


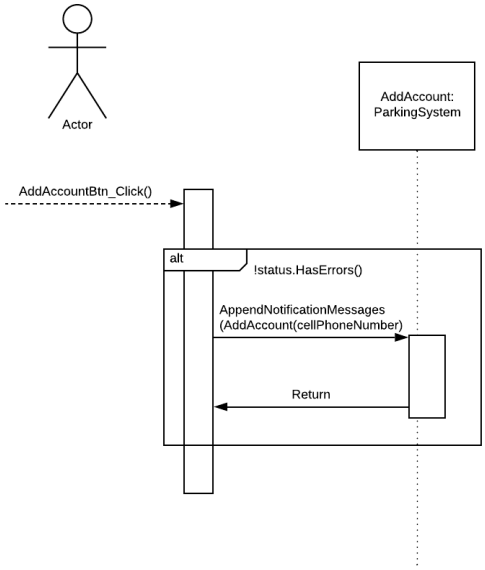
Diagrama de clases



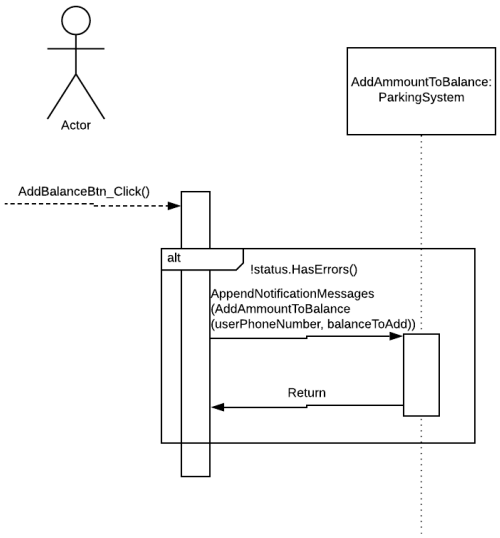
Diagramas de interacción

A continuación, se detallan los diagramas de interacción realizados para las funcionalidades de Agregar una cuenta, Agregar saldo, Realizar una reserva, Consulta de compras y Generación de reportes. Los siguientes diagramas se incluyen tanto en el repositorio como en el archivo entregado para una mejor visualización.

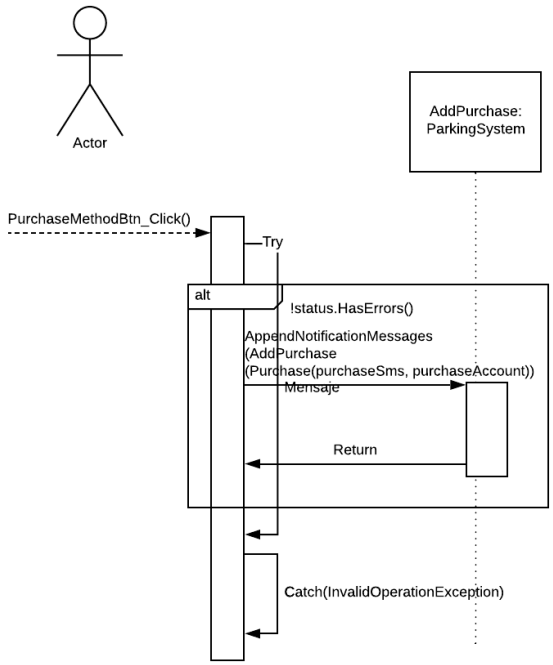
Agregar una cuenta



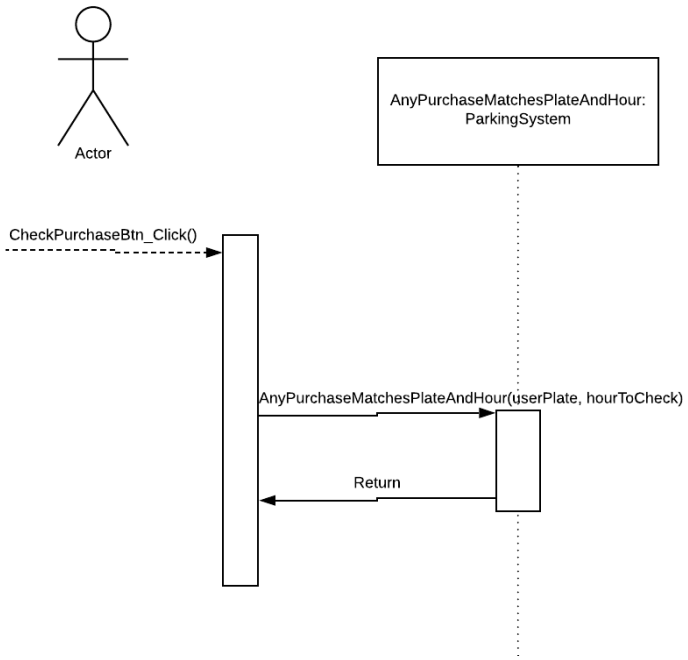
Agregar saldo



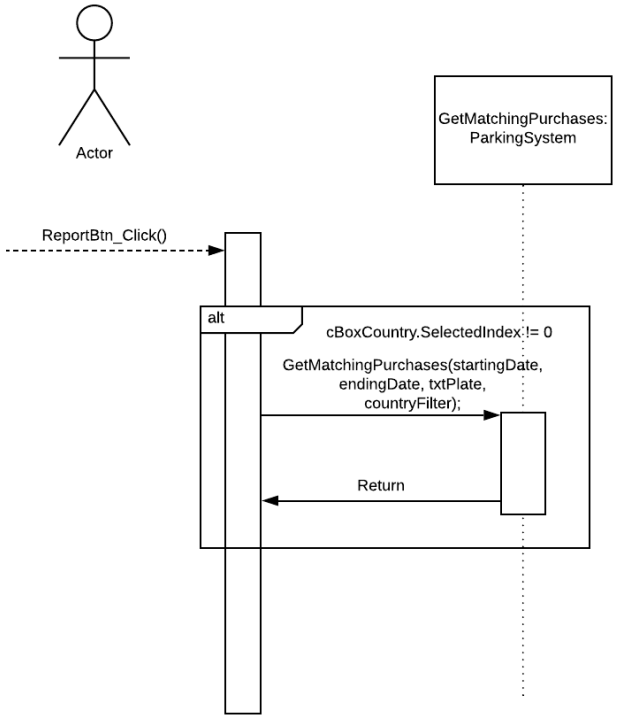
Realizar una reserva



Consulta por una reserva

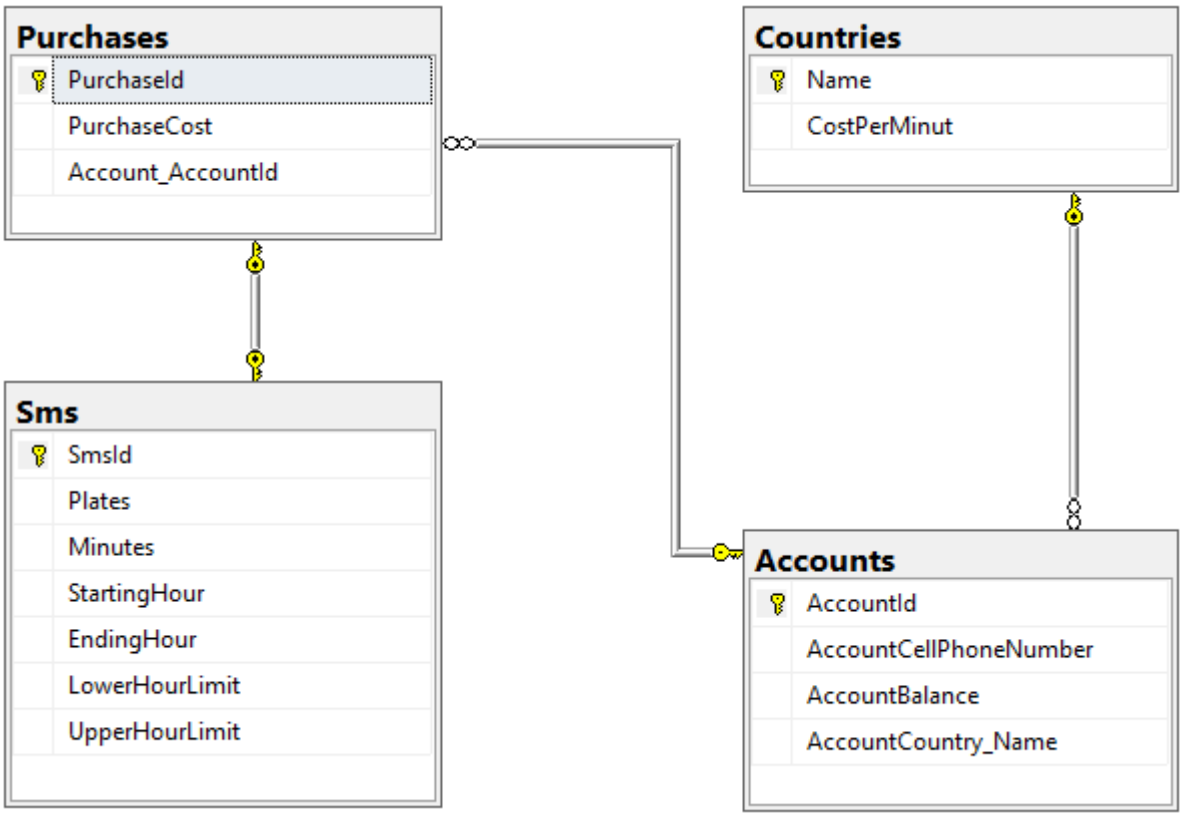


Generación de reportes



Modelo de tablas de la estructura de la base de datos

A continuación, se presenta el modelo de tablas de la estructura de la base de datos. Este es un Modelo Entidad Relación generado con SQL Server Managment Studio.



Cobertura de pruebas unitarias

La cobertura total de los paquetes BussinesLogic, Data, Entities y BussinesLogicTest es de un total de 95%. El cual se puede apreciar en las siguientes imágenes.

Jerarquía	No cubiertos (bloques)	No cubiertos (% de bloques)	Cubiertos (bloques)	Cubiertos (% de bloques)
joelp_DESKTOP-NHGTK34 2019-11-21 18_36_39.coverage	105	4,98 %	2004	95,02 %
businesslogic.dll	12	7,06 %	158	92,94 %
{ } BusinessLogic	12	7,06 %	158	92,94 %
ParkingSystem	8	5,13 %	148	94,87 %
ParkingSystem.<>c	0	0,00 %	4	100,00 %
ParkingSystemInitializer	4	40,00 %	6	60,00 %
businesslogictest.dll	21	2,01 %	1024	97,99 %
{ } BusinessLogicTest	21	2,01 %	1024	97,99 %
AccountTest	0	0,00 %	61	100,00 %
AccountTransactionValidatorTest	0	0,00 %	22	100,00 %
ArgCellPhoneValidatorTest	0	0,00 %	52	100,00 %
ArgSmsValidatorTest	12	14,81 %	69	85,19 %
BalanceValidatorTest	0	0,00 %	15	100,00 %
CountryTest	0	0,00 %	4	100,00 %
NotificationTest	1	1,61 %	61	98,39 %
OrmAccountRepositoryTest	0	0,00 %	41	100,00 %
OrmCountryRepositoryTest	0	0,00 %	36	100,00 %
OrmPurchaseRepositoryTest	0	0,00 %	119	100,00 %
ParkingSystemTest	2	1,18 %	168	98,82 %
ParkingSystemTest.<>c	0	0,00 %	3	100,00 %
PurchaseTest	0	0,00 %	22	100,00 %
RepositoryTest	0	0,00 %	133	100,00 %
RepositoryTest.<>c__DisplayClass15_0	2	50,00 %	2	50,00 %
RepositoryTest.TestClass	1	25,00 %	3	75,00 %
SmsTest	1	3,70 %	26	96,30 %
UyCellPhoneValidatorTest	0	0,00 %	36	100,00 %
UySmsValidatorTest	1	0,66 %	150	99,34 %
UySmsValidatorTest.<>c__DisplayClass25_0	1	50,00 %	1	50,00 %
data.dll	18	6,72 %	250	93,28 %
{ } Data	18	6,72 %	250	93,28 %
OrmAccountRepository	0	0,00 %	47	100,00 %
OrmCountryRepository	0	0,00 %	30	100,00 %
OrmPurchaseRepository	0	0,00 %	131	100,00 %
ParkingContext	1	8,33 %	11	91,67 %
Repository<T>	8	42,11 %	11	57,89 %
UnitOfWork	9	31,03 %	20	68,97 %
entities.dll	54	8,63 %	572	91,37 %

Jerarquía	No cubiertos (bloques)	No cubiertos (% de bloques)	Cubiertos (bloques)	Cubiertos (% de bloques)
joelp_DESKTOP-NHGTK34 2019-11-21 18_36_39.coverage	105	4,98 %	2004	95,02 %
businesslogic.dll	12	7,06 %	158	92,94 %
businesslogictest.dll	21	2,01 %	1024	97,99 %
data.dll	18	6,72 %	250	93,28 %
entities.dll	54	8,63 %	572	91,37 %
{ } Entities	54	8,63 %	572	91,37 %
Account	9	13,04 %	60	86,96 %
Account.<>c__DisplayClass28_0	0	0,00 %	5	100,00 %
AccountTransactionValidator	0	0,00 %	20	100,00 %
ArgCellPhoneValidator	12	30,77 %	27	69,23 %
ArgSmsValidator	2	4,00 %	48	96,00 %
BalanceValidator	0	0,00 %	18	100,00 %
CellPhoneValidator	0	0,00 %	16	100,00 %
Country	0	0,00 %	27	100,00 %
Notification	0	0,00 %	53	100,00 %
Purchase	18	40,00 %	27	60,00 %
Sms	9	11,84 %	67	88,16 %
Sms.<>c__DisplayClass39_0	0	0,00 %	5	100,00 %
SmsValidator	0	0,00 %	108	100,00 %
SmsValidator.<>c	0	0,00 %	2	100,00 %
SmsValidator.<>c__DisplayClass10_0	0	0,00 %	2	100,00 %
UyCellPhoneValidator	2	6,45 %	29	93,55 %
UySmsValidator	2	4,26 %	45	95,74 %
Validator	0	0,00 %	13	100,00 %