



MANUAL DE PROGRAMADOR

Tabla de contenido

Introducción	3
Presentación del Producto	3
Descripción General	4
Descripción Detallada de Requerimientos	4
Requerimientos de Interfaz	10
Estructura de base de datos	11
Clases	13
Funcionamiento del programa	17
Restricciones de diseño	25
Requerimientos de Licencia	25
Observaciones	25

1. Introducción

1.1. Objetivo

El presente Manual del programador tiene como objetivo definir con claridad los requerimientos correspondientes al proyecto League Planner, así como su diseño e implementación.

Este documento contiene toda aquella información necesaria para que el lector de este documento pueda entender los objetivos y el funcionamiento del producto mencionado

1.2. Definiciones, Acrónimos y Abreviaturas

Por ahora no hay ninguno.

1.3. Audiencia

Este documento está dirigido a las siguientes. Personas y Roles dentro del Proyecto:

- *Programadores del sistema*
- *Diseñadores del sistema*
- *Project manager del sistema*

1.4. Alcance

El alcance de la ERS comprende la definición de los requerimientos funcionales y no funcionales, como también otros aspectos que definen el producto, incluyendo objetivo del producto, restricciones, lo que el sistema no contempla, reglas de negocio, requerimientos de interfaz, restricciones de diseño, requerimientos de licencia o componentes comprados necesarios para el producto a desarrollarse, entre otras cosas.

2. Presentación del Producto

2.1. Propósito del Sistema

Objetivo

El software league planner se planteó para solucionar los múltiples problemas que implica la administración de una liga deportiva a nivel local ya que en la actualidad esta actividad se realiza de manera "manual" o con herramientas computacionales poco adecuadas la mayoría de este tipo de ligas con todos los contratiempos y desperdicio de recursos que esto implica.

2.2. Alcance

Todo aquel usuario que lidere, organice y dirija a un equipo de futbol soccer. ofreciendo una mayor facilidad y eficiencia comparado al método convencional con que el que se cuenta.

El Sistema no contempla

- *Versión para los dispositivos móviles (Temporalmente).*
- *Existencia de más de un usuario que controle los movimientos y la organización del equipo.*
- *Compatibilidad con MacOS.*

2.2. Restricciones y Supuestos

- *El programa será compatible solamente con Windows.*
- *El usuario que no sea administrador solo podrá hacer consultas de los resultados.*

3. Descripción General

3.1. Contexto del Producto

El producto funcionará sobre un sistema operativo windows y utilizara para el desarrollo de interfaces windows forms o WPF .

Perspectivas futuras del producto

Desarrollar una aplicación móvil y una plataforma web para que los involucrados puedan consultar la información de la liga en cualquier lugar.

3.2. Reglas y Funciones de Negocio

Ninguna por el momento

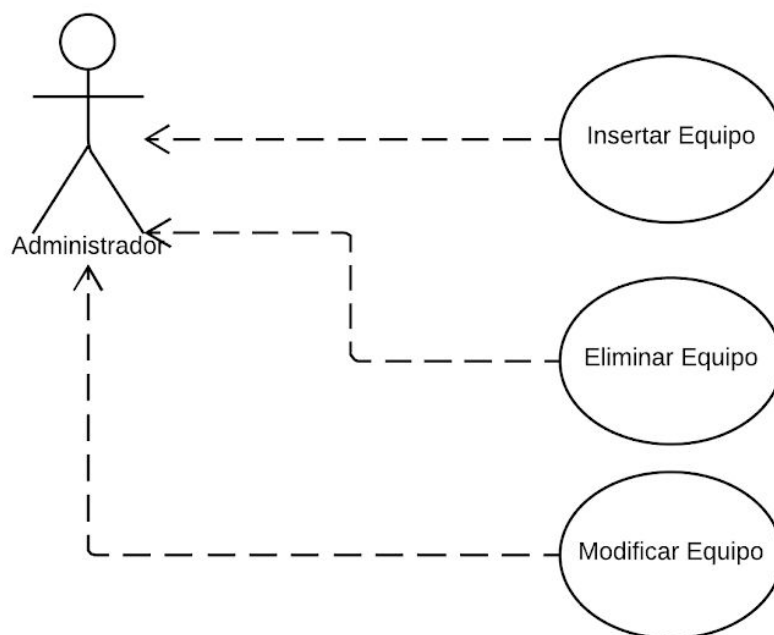
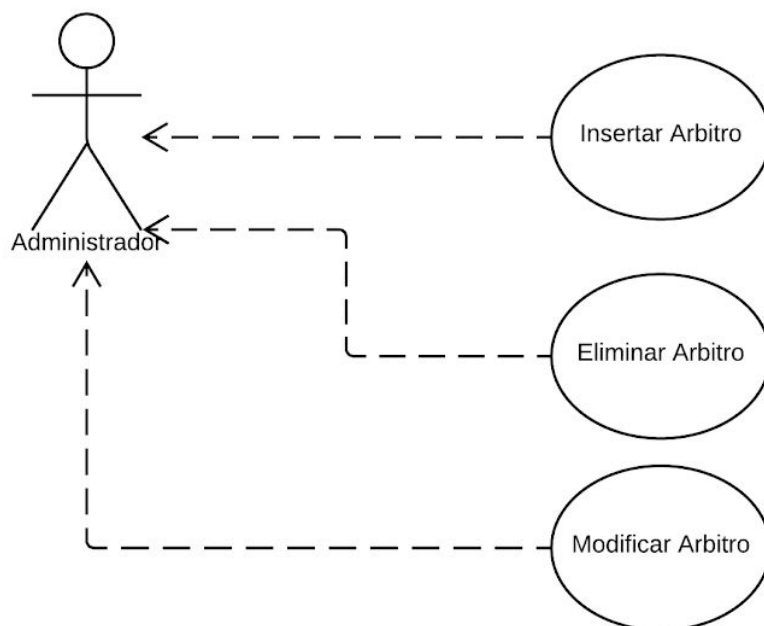
4. Descripción Detallada de Requerimientos

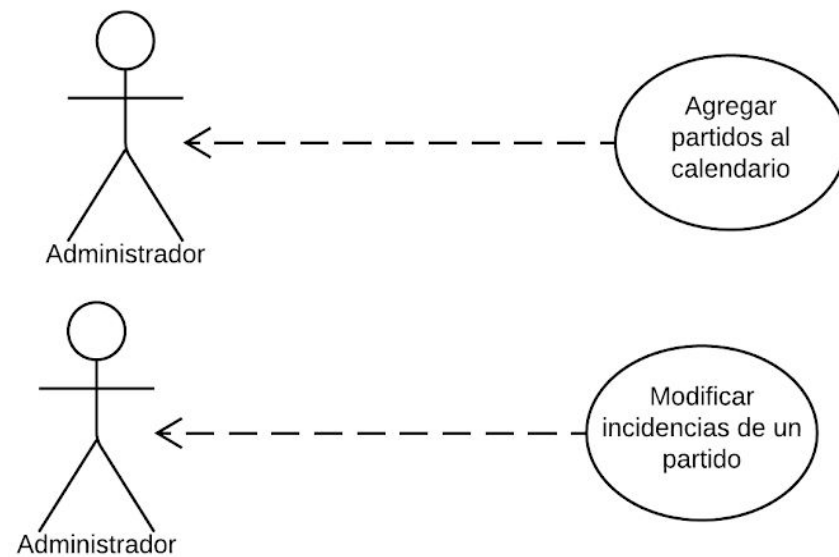
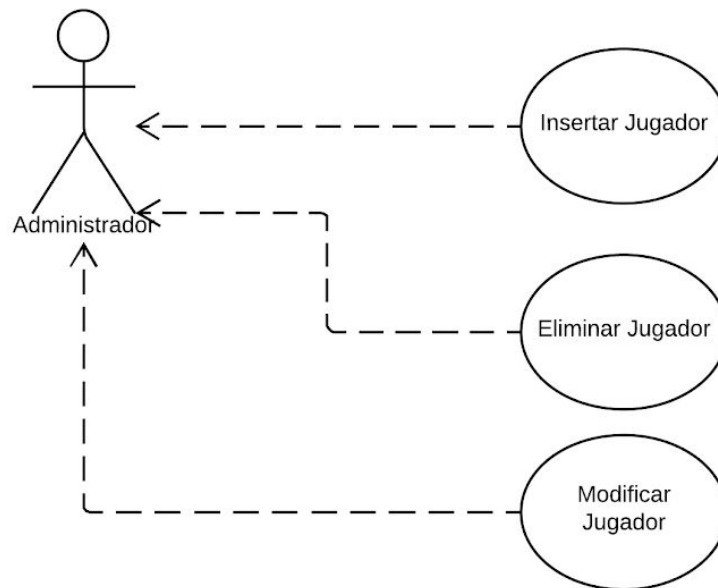
4.1. Actores

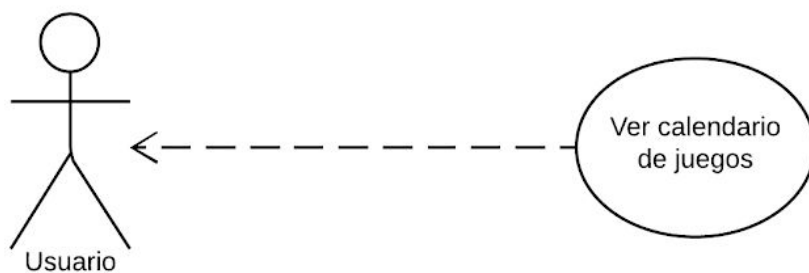
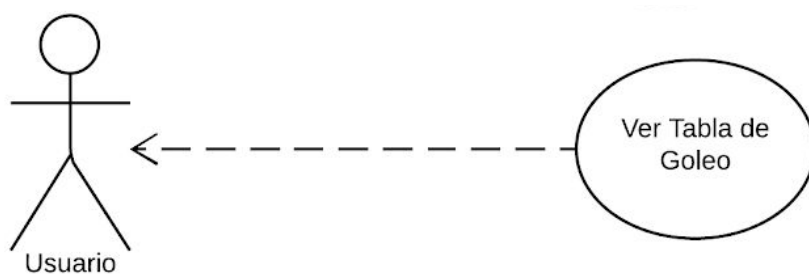
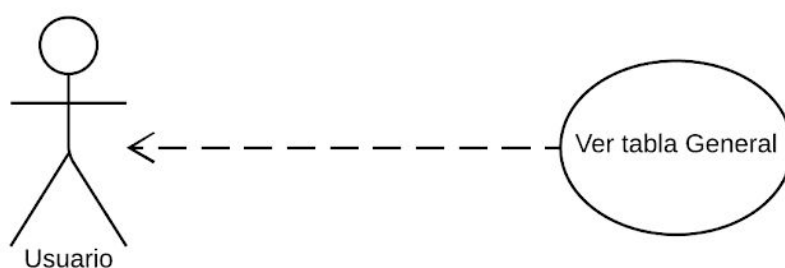
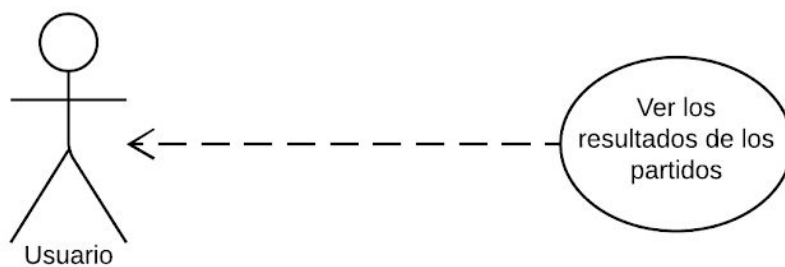
Administrador del sistema: *Es la persona que además de poder hacer las mismas acciones que el usuario podrá modificar la configuración del sistema, dar de alta y baja a los jugadores, equipos, árbitros y modificar resultados de los partidos.*

Usuario del sistema: *Es la persona que se encargará de consultar la información en el sistema que es relevante para él, como resultados de los partidos, tabla general y de goleadores*

Diagramas de Casos de Uso







4.2. Requerimientos Funcionales

- *Contar con un archivo donde se podrán guardar los datos de los jugadores.*
- *Definición de quién podrá entrar al sistema.*
- *Definición de quién podrá ingresar datos al sistema.*
- *El sistema deberá permitir hacer cambios por el administrador.*
- *El sistema será utilizado en computadoras con Windows.*
- *El sistema deberá tener soporte para actualizaciones.*

4.3. Requerimientos No Funcionales

- *Poder controlar las altas y bajas de los jugadores*
- *Tener una visualización de los resultados de los partidos*
- *Contar con un calendario que notifique los próximos eventos*
- *Saber la información de un jugador realizada en los partidos (goles anotados, lesiones, faltas asistencia, posición).*
-

4.3.1. Del Producto

Usabilidad:

RN-1 *El sistema está preparado para ser operado a través de mouse y teclado o panel táctil.*

RN-2 *Las pantallas serán desarrolladas para ambiente Windows 7 ó posterior, con resolución de pantalla de 1280 x 720, o superior.*

RN-3 *Todas las pantallas deben tener un modo de cancelar la operación en curso.*

RN-4 *Desplegar mensajes de error y advertencia intuitivos.*

RN-5 *Verificar/validar límites de campos y tipos de datos de las pantallas en relación al modelo de datos.*

Confiabilidad:

RN-7 *Implementar mecanismos que aseguren la integridad de los datos.*

Performance:

RN-9 *Se espera que el tiempo de respuesta en el momento de presionar un botón para continuar con el flujo de la información que no supere los 20 segundos.*

RN-10 *El sistema deberá liberar a todos los recursos de memoria al momento de cerrar una ventana y finalizar una funcionalidad.*

Soportabilidad:

RN-11 *El control de integridad de datos se hará del lado de la capa de datos (a nivel de la base de datos utilizando las claves foráneas). Los mensajes de error serán capturados por la aplicación y serán visualizados al usuario final.*

Documentación:

RN-13 *Correcta redacción y ortografía en las pantallas.*

RN-14 *Uso estandarizado de pantallas, mensajes y estilos.*

4.3.2. Del Ambiente

Ético:

RN-16 *El sistema debe garantizar la confidencialidad de la información de los Clientes y de los valores negociados con el Cliente.*

5. Requerimientos de Interfaz

5.1. Interfaces de Hardware .

El sistema no contempla el uso de interfaces de hardware.

5.2. Interfaces de Software

El sistema hace uso de SQLite para el manejo de los datos ()

5.3. Interfaces de Comunicación

El sistema no contempla el uso de interfaces de comunicación

6. Estructura de base de datos

Tabla árbitros








 id	INTEGER
 nombre	varchar (50)
 apellidop	varchar (50)
 apellidom	varchar (50)
 nacimiento	TEXT
 telefono	varchar (10)
 email	varchar (100)
 eliminado	int

Tabla calendarios


 id	INTEGER
 fecha	DateTime
 local	varchar (30)
 visitante	varchar (30)
 ganador	INTEGER

Tabla entrenadores







 id	INTEGER
 nombre	varchar (50)
 apellidop	varchar (50)
 apellidom	varchar (50)
 nacimiento	TEXT
 telefono	int
 email	varchar (100)
 eliminado	int

Tabla equipos





 id	INTEGER
 nombre	varchar (50)
 clave_entrenador	int
 eliminado	int
 activo	int

Tabla goles






 id	INTEGER
 clave_jugador	int
 clave_partido	int
 favor_o_contra	int
 clave_equipo	int

Tabla jugadores














 id	INTEGER
 nombre	varchar (50)
 apellidop	varchar (50)
 apellidom	varchar (50)
 nacimiento	TEXT
 telefono	long
 email	varchar (100)
 clave_equipo	int
 eliminado	int

Tabla tarjetas

 id	INTEGER
 clave_jugador	int
 tipo	varchar (10)
 clave_partido	int
 clave_equipo	int

7. Clases

7.1 Clases modelo

Clases que representan el modelo de datos del sistema

Clase Tarjeta

Clase que se encarga de guardar los datos de cada tarjeta tales como: id, clave del jugador sancionado, clave del partido en que fue sancionado, tipo de tarjeta y el equipo al que se cuenta la tarjeta.

Clase Jugador

Clase Que sirve para guardar los datos de un jugador como: id, Nombre, apellido materno, apellido paterno, fecha de nacimiento, teléfono, e-mail clave del equipo al que pertenece

Clase Gol

Clase que se encarga de guardar los datos de cada gol tales como: id, clave del jugador que lo anotó, clave del partido en que fue anotado y el equipo al que se cuenta el gol.

Clase Equipo

Se encarga de guardar los datos de un equipo como nombre, y id

Clase Calendario

Se encarga de obtener la fecha y los equipos que estarán jugando en dicha fecha.

Clase Árbitro

Clase Que sirve para guardar los datos de un jugador como: id, Nombre, apellido materno, apellido paterno, fecha de nacimiento, teléfono y e-mail.

7.2 Clases Controlador

Estas clases son usadas para interactuar con el modelo de datos y la aplicación y sirven como puente entre la base de datos y los objetos usados por la aplicación.

Clase ArbitroDBController

Se encarga de hacer las transacciones de inserción, actualización, eliminación y lectura de la tabla árbitros, funciona como un puente entre el tipo de datos Arbitro de la aplicación y la base de datos.

Clase CalendarioDBController

Se encarga de hacer las transacciones de inserción, actualización, eliminación y lectura de la tabla calendarios, funciona como un puente entre el tipo de datos Calendario de la aplicación y la base de datos.

Clase EquipoDBController

Se encarga de hacer las transacciones de inserción, actualización, eliminación y lectura de la tabla equipos, funciona como un puente entre el tipo de datos Equipo de la aplicación y la base de datos.

Clase GolDBController

Se encarga de hacer las transacciones de inserción, actualización, eliminación y lectura de la tabla goles, funciona como un puente entre el tipo de datos Gol de la aplicación y la base de datos.

Clase JugadorDBController

Se encarga de hacer las transacciones de inserción, actualización, eliminación y lectura de la tabla jugadores, funciona como un puente entre el tipo de datos Jugador de la aplicación y la base de datos.

Clase TarjetaDBController

Se encarga de hacer las transacciones de inserción, actualización, eliminación y lectura de la tabla tarjetas, funciona como un puente entre el tipo de datos Tarjeta de la aplicación y la base de datos.

7.3 Clases Vista

Estas clases ayudan a representar de manera visual en la aplicación el estado de los datos en determinado momento y a hacerlos comprensibles para el usuario en las distintas pantallas de la aplicación.

Clase LoginWindow.xaml.cs

Clase que sirve para desplegar la pantalla de login de la aplicación, la cual muestra dos campos de texto: usuario y contraseña y un botón para ingresar a la aplicación con las credenciales.

Clase AdminMainWindow.xaml.cs

Clase que sirve para desplegar el menú principal del administrador la cual se divide en 3 partes principales Título y logo en la parte superior, el panel de opciones en la parte central y las opciones de sesión en la parte inferior izquierda.

Clase ArbitrosCatalogo.xaml.cs

Clase que sirve para Mostrar y administrar todos los árbitros, en ella podemos visualizar los datos y navegar entre cada uno de ellos, también podemos buscar, agregar eliminar y editar.

Clase ArbitroPage.xaml.cs

Clase que sirve para que podamos editar los datos de un solo árbitro, ya sea uno existente o nuevo, cuenta con cada uno de los campos para los datos del árbitro, un botón para aceptar los cambios y un botón para cancelar.

Clase EquiposCatalogo.xaml.cs

Clase que sirve para Mostrar y administrar todos los equipos, en ella podemos visualizar los datos y navegar entre cada uno de ellos, también podemos buscar, agregar eliminar y editar.

Clase EquipoPage.xaml.cs

Clase que sirve para que podamos editar los datos de un solo equipo, ya sea uno existente o nuevo, cuenta con cada uno de los campos para los datos del equipo, un botón para aceptar los cambios y un botón para cancelar.

Clase JugadoresCatalogo.xaml.cs

Clase que sirve para Mostrar y administrar todos los jugadores, en ella podemos visualizar los datos y navegar entre cada uno de ellos, también podemos buscar, agregar eliminar y editar.

Clase JugadorPage.xaml.cs

Clase que sirve para que podamos editar los datos de un solo jugador, ya sea uno existente o nuevo, cuenta con cada uno de los campos para los datos del jugador, un botón para aceptar los cambios y un botón para cancelar.

Clase CalendarioCatalogo.xaml.cs

Clase que sirve para mostrar al administrador los partidos y permitirle agregar nuevos, consta de una lista en la que podemos ver los detalles mas relevantes de cada uno de los partidos y un boton para agregar partidos.

Clase CalendarioPage.xaml.cs

Clase que sirve para editar los datos de un evento nuevo que se va a agregar, tiene campos para seleccionar la fecha y la hora del partido, así como campos para seleccionar al equipo visitante y local.

Clase CalendarioViewPage.xaml.cs

Clase que sirve para mostrar al usuario de manera amigable los partidos que el administrador ha agregado al calendario, consta de una tabla que muestra los detalles del partido programado.

Clase IncidenciasPage.xaml.cs

Clase que muestra y permite modificar las incidencias ocurridas en los partidos, como goles o tarjetas, cuenta con un campo para seleccionar el partido en la parte superior, campos para seleccionar el jugador que ha anotado determinado gol para cada equipo, campos para seleccionar jugadores con distintos tipos de tarjetas y una lista que muestra las incidencias asociadas a cada equipo para el partido, en la parte inferior el botón aceptar para guardar los cambios.

Clase UsuarioMainWindow.xaml.cs

Esta clase sirve para mostrar un menú con la misma estructura que el de administrador pero con menos opciones ya que es para el usuario y solo permite visualizar datos y no permite modificar ni insertar datos.

Clase ResultsPage.xaml.cs

Esta clase Sirve para mostrar al usuario de manera amigable los resultados de los partidos, consta de un encabezado y una tabla en la que se detallan los resultados de los partidos así como sus incidencias.

Clase GoleadoresPage.xaml.cs

Clase que muestra La tabla de goleo del torneo, se compone de un título y de una tabla que muestra a los jugadores con más goles ordenados de forma ascendente, desde el mejor goleador hasta el peor.

Clase TablaPage.xaml.cs

Esta clase muestra la tabla general del torneo, consta de un título y una tabla en la que se muestra de manera detallada las estadísticas de cada equipo como partidos jugados, ganados, empatados, perdidos, goles a favor, goles en contra, diferencia de goles y puntos, los equipos están ordenados de mayor cantidad de puntos a menor cantidad de puntos.

8. Funcionamiento del programa

Herramientas necesarias:

- Visual studio 2017
- SQLite para Visual Studio 2017

Instalación de las herramientas

Instalación de Visual Studio 2017

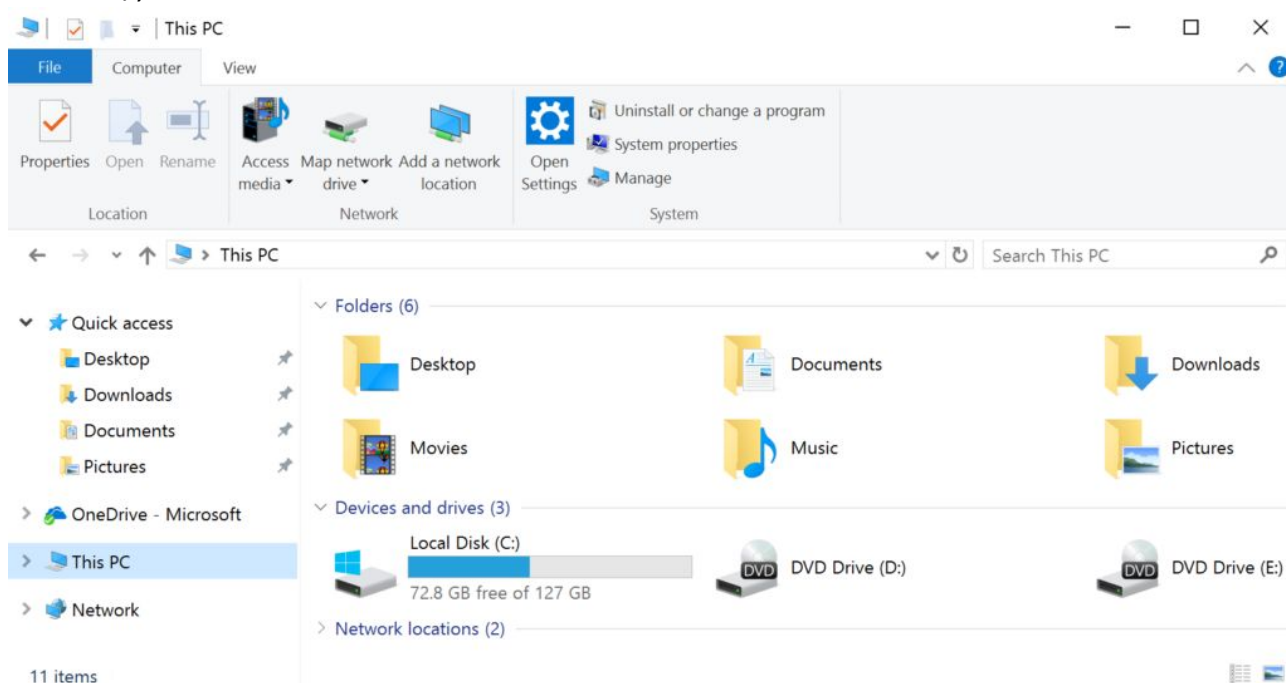
Paso 1: antes de la instalación

Vamos a comprobar que nuestra máquina está lista para que instalemos Visual Studio 2017. Tendremos que realizar algunas actualizaciones y liberar algo de espacio.

[✓ Aplique las actualizaciones de Windows más recientes.](#)

[✓ Compruebe que el espacio en disco sea suficiente.](#)

- Abra “Mi PC” en el explorador de archivos.
- Asegúrese de tener más de 7 GB de espacio disponible en una unidad (se recomienda la unidad C:/).

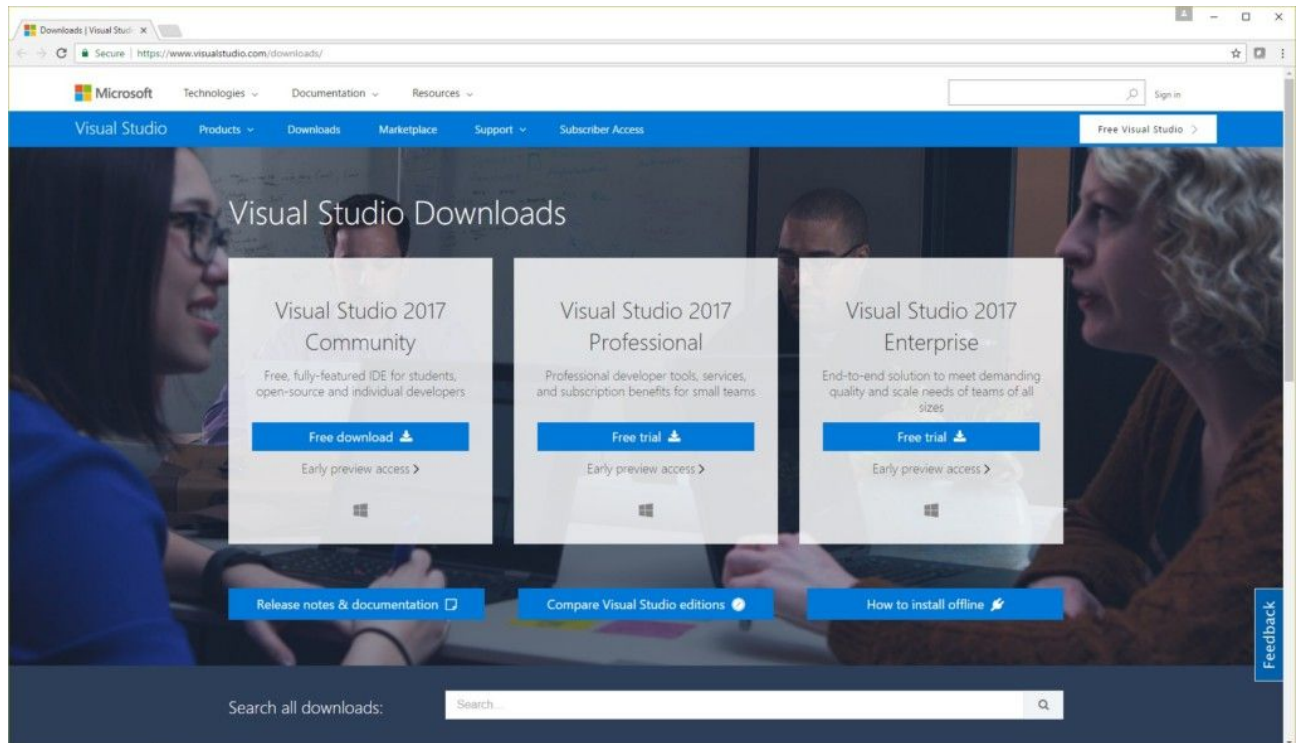


Paso 2: descarga

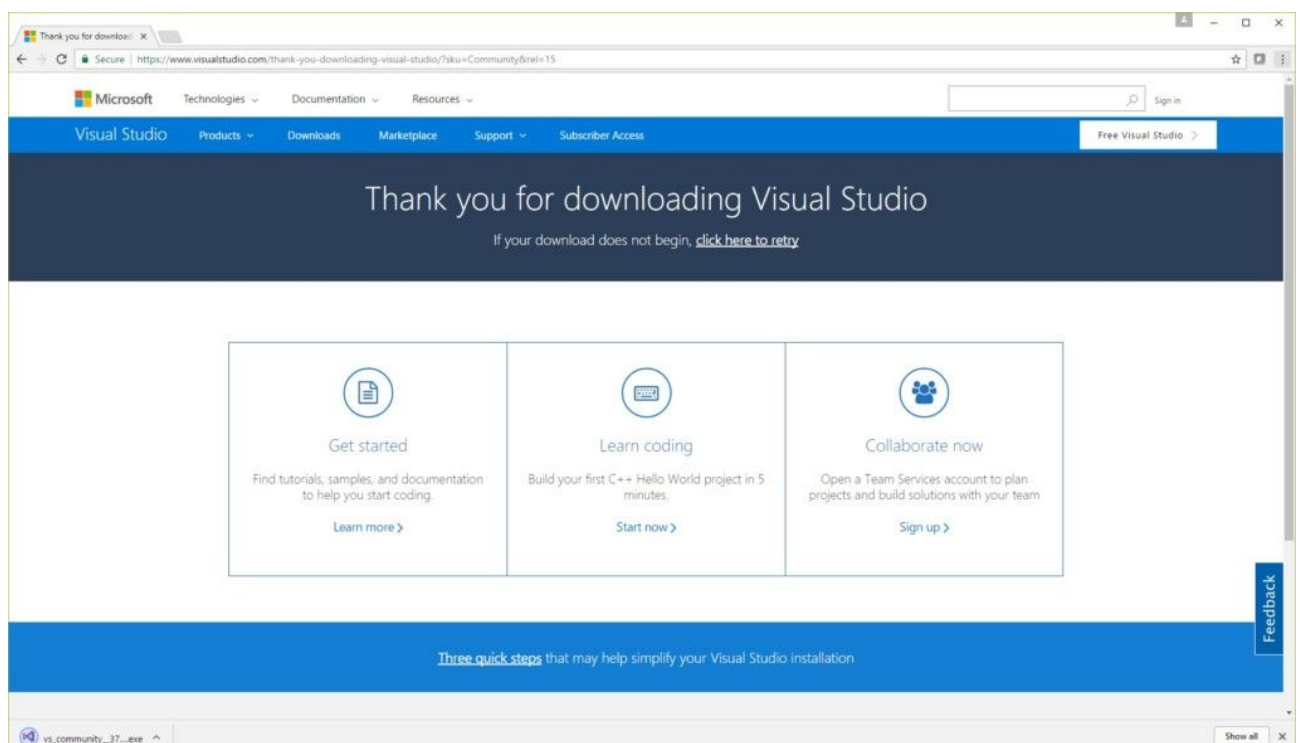
Ahora, vamos a obtener nuestra descarga de Visual Studio 2017 para prepararnos para la instalación.

✓ Descargue Visual Studio 2017.

- Vaya a [VisualStudio.com/Downloads](https://visualstudio.com/Downloads) y seleccione su versión



✓ El archivo se descargará y deberá usar “Guardar como” si es necesario.

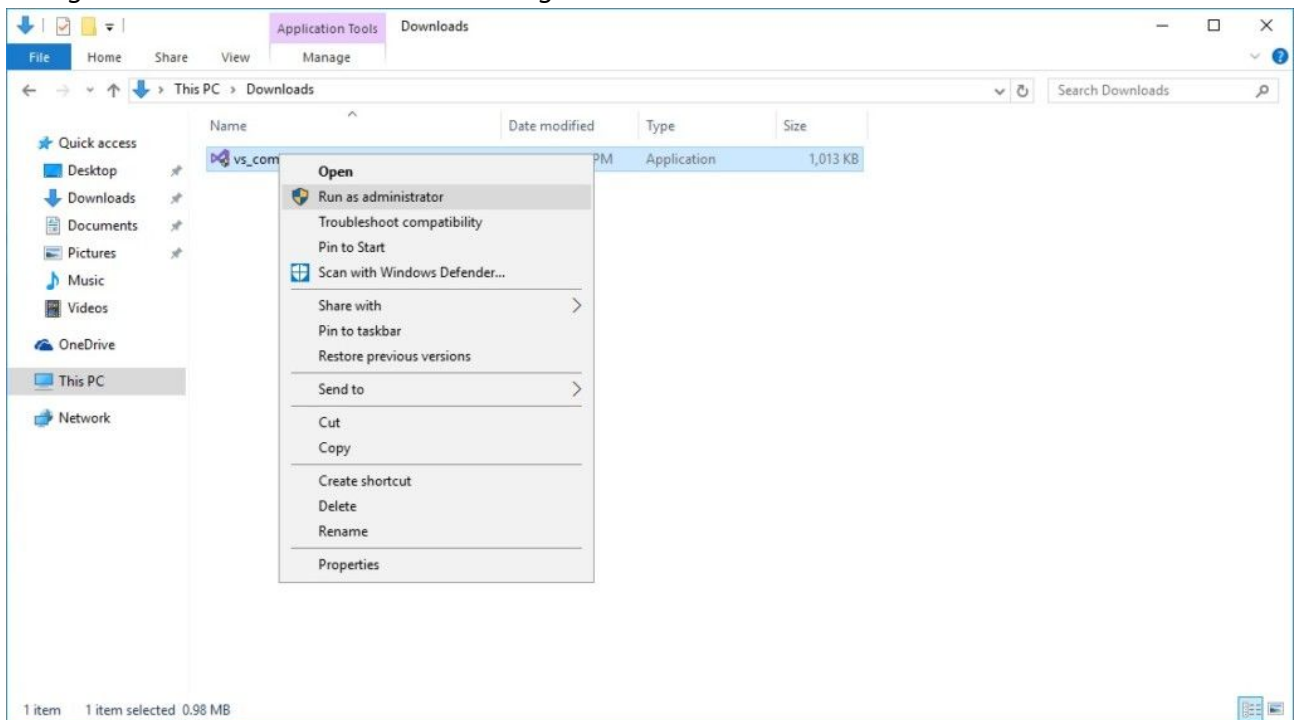


Paso 3: instalación

Seleccione los componentes que quiere instalar o use la opción de “instalación predeterminada” para obtener las herramientas más comunes (esto será suficiente para este caso) Para obtener más información sobre los distintos componentes, consulte *Instalación de Visual Studio 2017: Instalar cargas de trabajo*.

✓ Localice el archivo, haga clic con el botón derecho y seleccione “Ejecutar como administrador”.

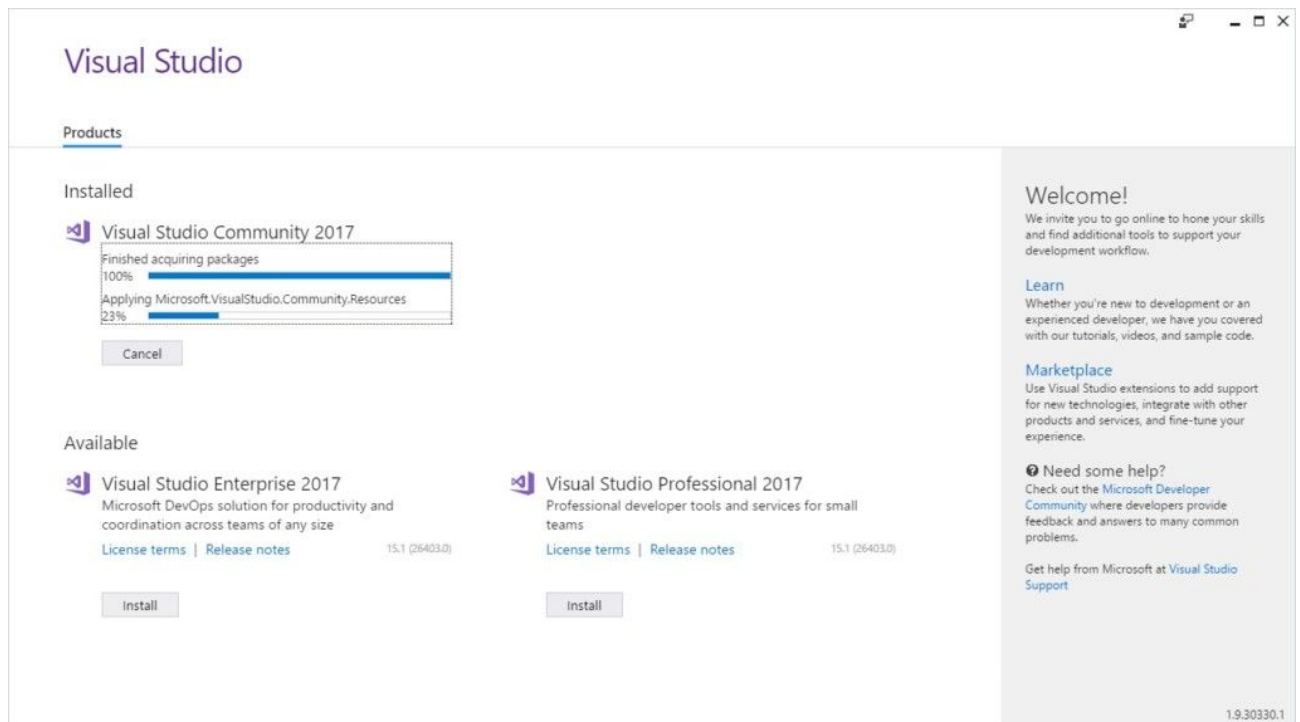
- Navegue hasta la ubicación donde descargó el archivo .exe.



✓ Haga clic para ejecutarlo.



✓ Realice la instalación y espere hasta que finalice.

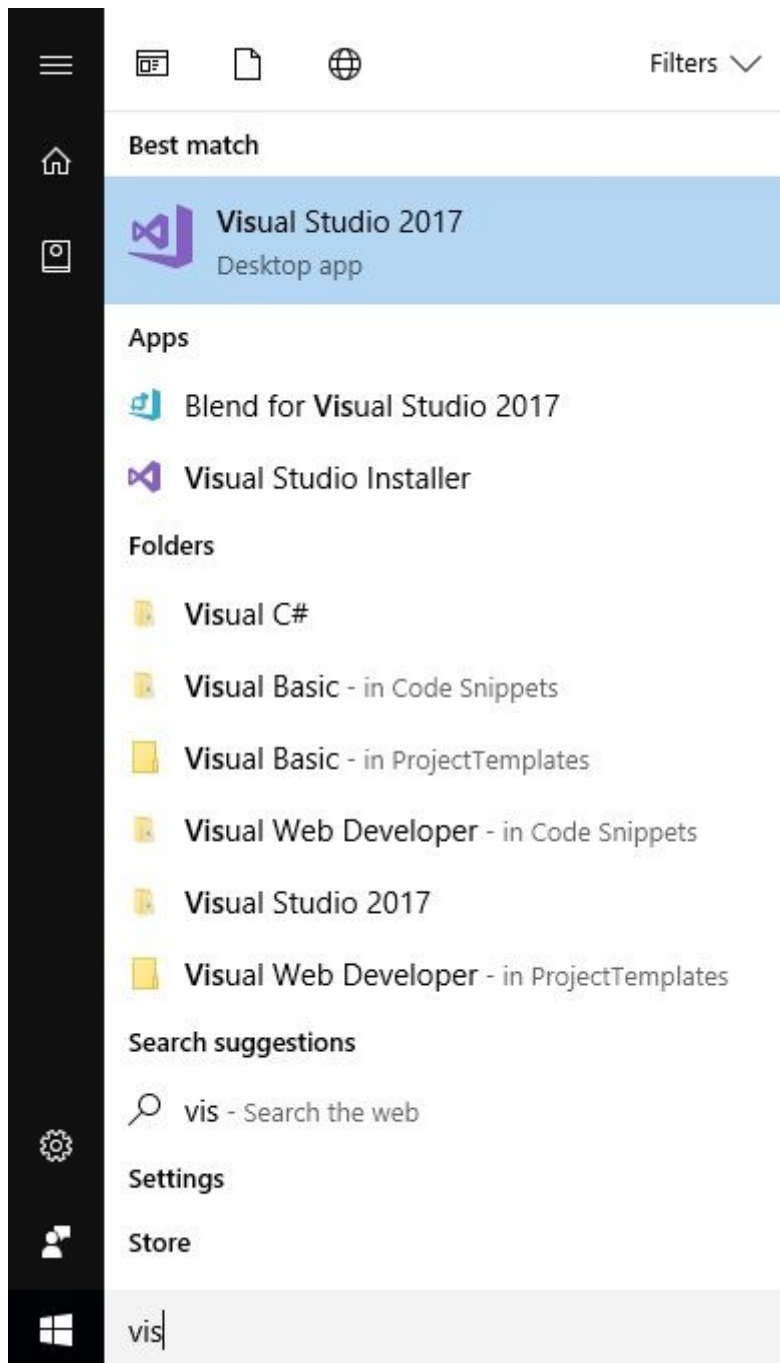


Paso 4: inicio

Ya está listo para la acción.

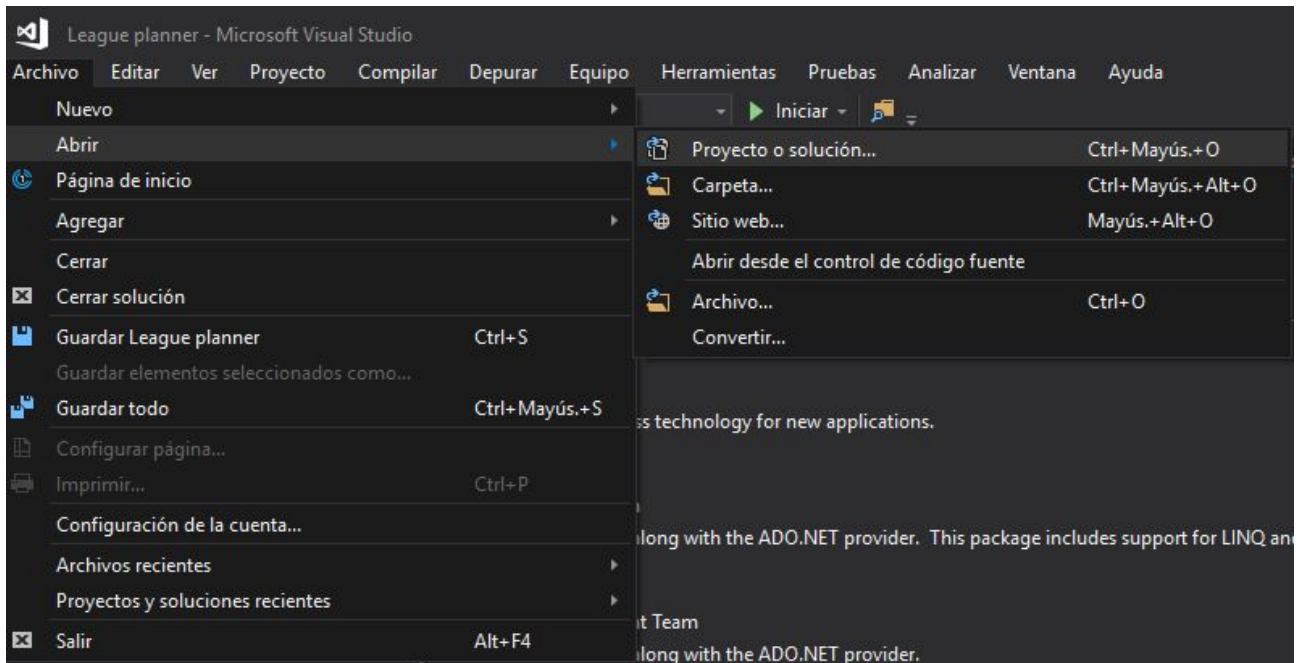
✓ Si es necesario, reinicie el equipo.

✓ Inicie Visual Studio 2017.

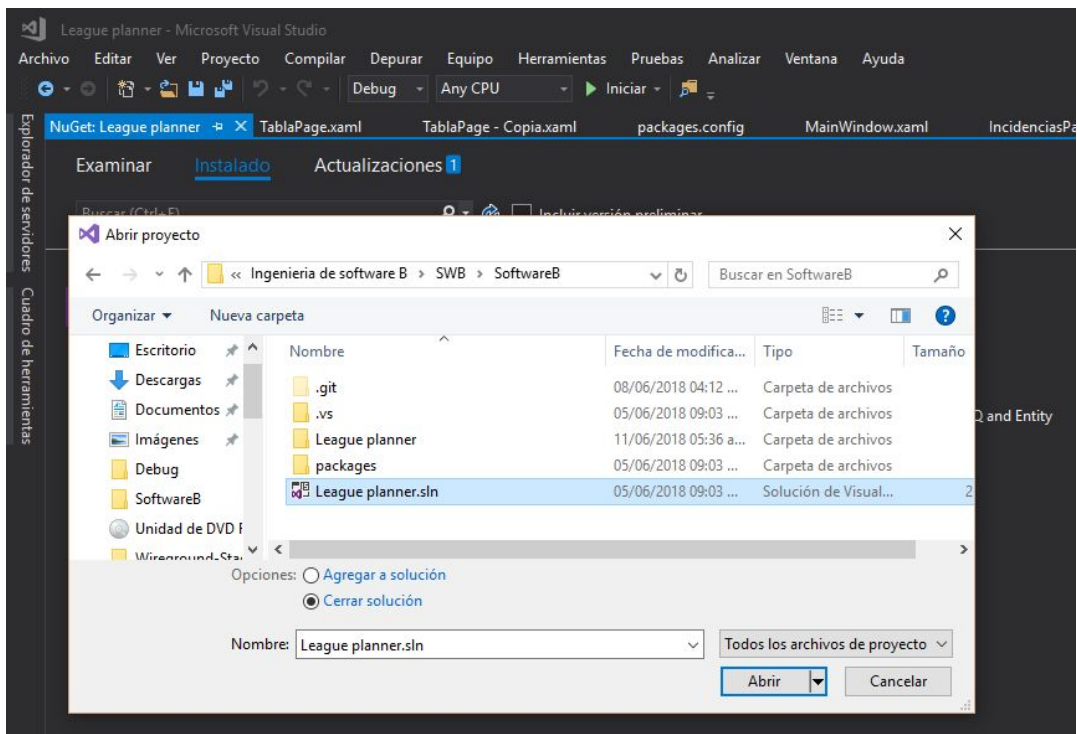


Paso 5: abrir proyecto

Vaya al menú Archivo en la parte superior izquierda, seleccione el submenú Abrir y a continuación haga clic en Proyecto o solución...



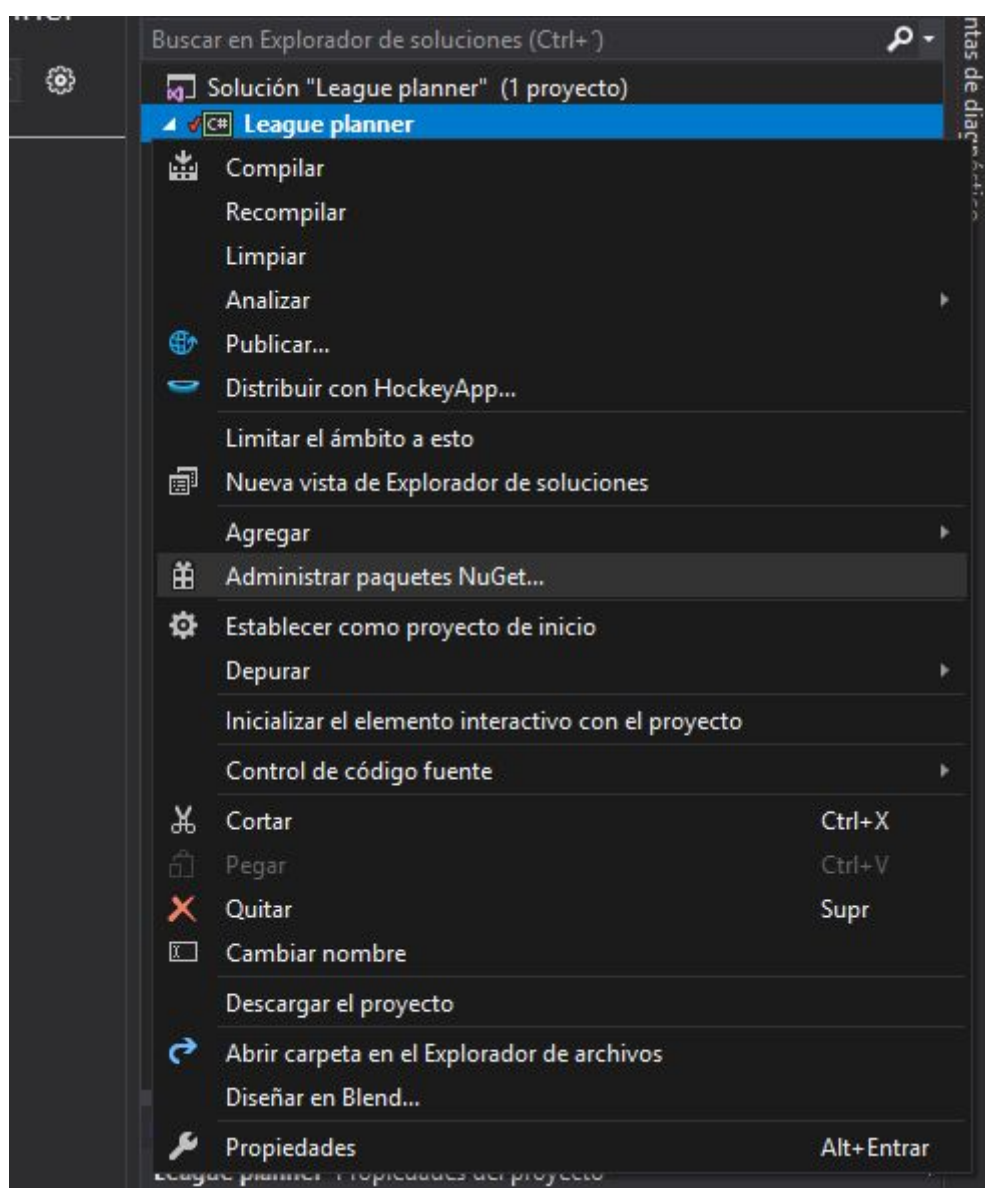
- En el explorador de archivos emergente localice y seleccione el archivo *League planner.sln* que esta en la carpeta del código fuente proporcionada.



Instalación de SQLite para Visual Studio

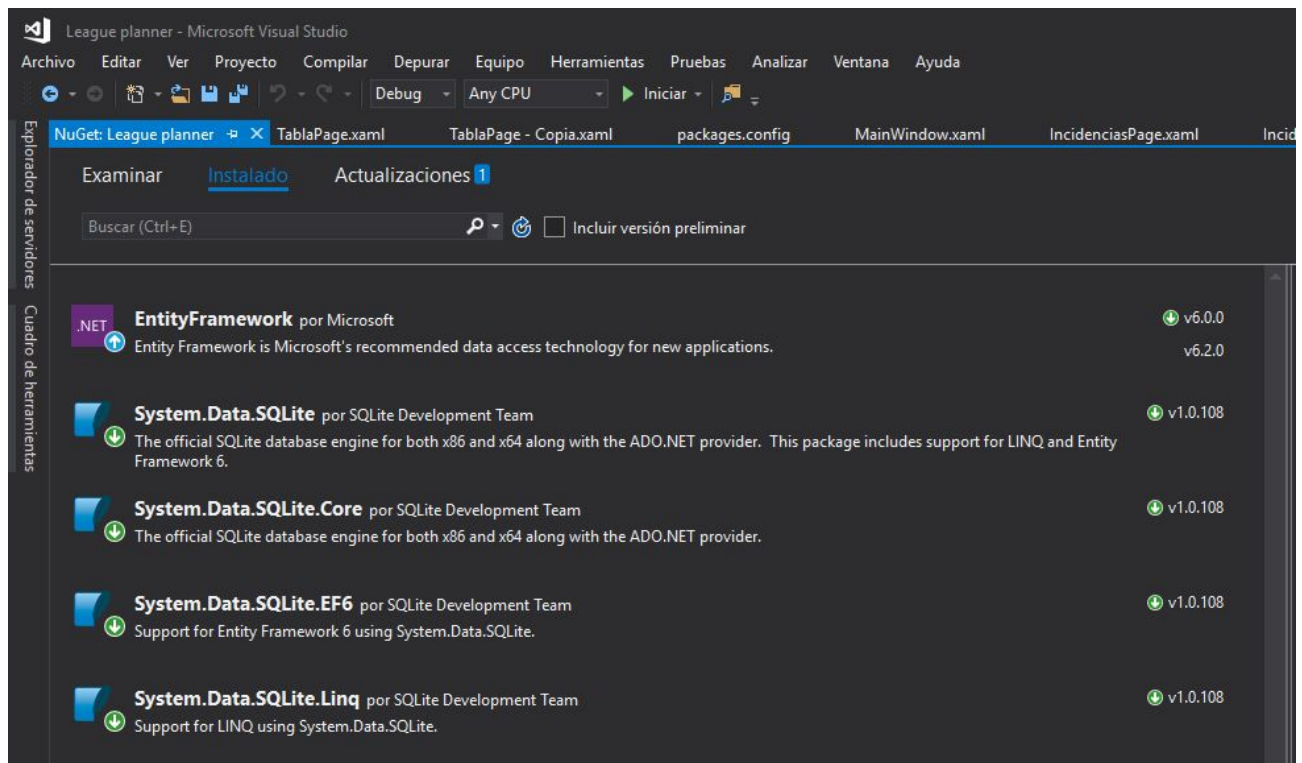
Paso 1: Abrir gestor de paquetes NuGet

Con el proyecto abierto nos posicionamos sobre la sección Explorador de soluciones y damos clic derecho sobre el proyecto League planner, aparecerá un menú, damos click sobre Administrar paquetes NuGet.



Paso 2: Seleccionar e instalar paquetes

Aparecer la siguiente ventana, en la sección Examinar busque e instale los paquetes que se muestran a continuación si no se encuentran instalados.



9. Restricciones de diseño

Solo las restricciones inherentes al sistema operativo y herramientas usadas para el desarrollo

10.Requerimientos de Licencia

Se requiere una licencia de Visual Studio Enterprise.