

SISTEMA PARA LA ANOTACIÓN DE JUEGOS DE BÉISBOL

ANNOTATION SYSTEM FOR BASEBALL GAME

Ing. Daniel Eduardo Guerrero Ramírez, dguerrero@grm.uci.cu

Facultad Regional de Granma de la Universidad de las Ciencias Informáticas

Ing. Ramón Ernesto Stevenson Borrell, restevenson@hab.uci.cu

Facultad Regional de Artemisa de la Universidad de las Ciencias Informáticas

Ministerio de la Informática y las Comunicaciones

Manzanillo, Granma

Julio de 2011

“Año 53 de la Revolución”

RESUMEN

En Cuba existe actualmente un sistema de anotación de juegos de béisbol, sin embargo el mismo no cumple con las necesidades y exigencias actuales de este deporte a nivel internacional. Con el objetivo de mejorar la calidad, eficiencia y precisión en la anotación de los juegos de béisbol desarrollados en Cuba, surge la idea de crear un sistema que cumpla con los requisitos necesarios para realizar esta tarea de una forma más eficiente y amena.

El nuevo sistema cuenta con nuevas funcionalidades y una interfaz de usuario amigable permitiendo una mejor interacción entre el operador y el sistema, además de cumple con los requisitos y exigencias planteadas por los usuarios a los cuales va dirigido el mismo.

Este trabajo constituye todo el proceso de desarrollo de software realizado para la obtención de un producto final, y en él se justifica y argumenta la elección y utilización de cada una de las herramientas y técnicas de desarrollo utilizadas durante la concepción del mismo.

Palabras Clave: Sistema, anotación, requisitos, operador, software.

ABSTRACT

In Cuba there is now a baseball games scoring system, but it does not meets the current needs and demands of the sport internationally. With the aim of improve quality, efficiency and accuracy in recording baseball games developed in Cuba, comes abroad the idea of creating a system that meets the requirements necessary to perform this task in a more efficient and enjoyable way.

The new system includes new functionality and a friendly user interface allowing better interaction between the operator and the system, and meets the requirements and demands raised by users to whom it is addressed.

This work constitutes the entire software development process undertaken to obtain a final product, also in this product are justified and argued the choice and use of each of the tools and development techniques used for the design of it.

KeyWords: System, annotation, requirements, operator, software.

ÍNDICE

INTRODUCCIÓN	4
DESARROLLO	6
Análisis del sistema actual	6
Descripción General del Problema	7
Argumentación del problema a resolver	7
Metodologías	9
Proceso Unificado de Software (RUP), como metodología utilizada en el desarrollo del trabajo.	9
Plataforma .NET	11
Plataforma .NET y Mono	11
XML	13
Herramientas utilizadas	14
Lenguaje de programación	15
Problema y situación problemática	17
Propuesta del sistema	18
CONCLUSIONES	19
REFERENCIAS BIBLIOGRÁFICAS	20
BIBLIOGRAFÍA	20
ANEXOS	21

INTRODUCCIÓN

En Cuba el béisbol es pasión, pues es el deporte nacional. A lo largo de los años se han obtenido muchísimos logros en este deporte, lo que ha llevado a reconocer a Cuba como una de las principales potencias del béisbol amateur en el mundo, a pesar de ser un país con poco desarrollo económico. El Instituto Nacional de Educación Física y Recreación (INDER) y la Federación Cubana de Béisbol Amateur (FCBA) han sido protagonistas de los logros alcanzados por este deporte. La FCBA es la encargada de realizar numerosos procesos y tareas que se cumplen durante un juego de béisbol. La anotación de los juegos es uno de estos procesos.

Desde el año 1977 la FCBA puso en práctica un sistema de anotación de juegos de béisbol denominado “Béisbol”, el cual resulta de gran importancia, pues en él se recopilan una serie de datos y estadísticas necesarias para el desarrollo de este deporte. Sin embargo, con el desarrollo de las nuevas tecnologías surge también la necesidad de crear nuevos sistemas que sean capaces de recopilar una mayor cantidad de datos e información, que puedan ser almacenados para su posterior utilización, deviniendo en un uso óptimo de la información generada ya que permite brindar un grupo de estadísticas del juego en tiempo real y disponibles en cualquier momento.

De esta manera surge la idea de crear un sistema, que aparte de cumplir con todas las funcionalidades requeridas por la FCBA sea multiplataforma, debido a la tendencia actual que tiene el país hacia la migración al Software Libre, puesto que no está sujeto a las leyes del bloqueo económico impuesto por los Estados Unidos.

El objetivo general del trabajo es realizar la implementación de un sistema de anotación de juegos para el béisbol en Cuba. Se cuenta además con varios objetivos específicos los que permitirán darle cumplimiento al objetivo general, estos son los siguientes:

- Investigar otros sistemas de anotación existentes en Cuba y en otros países del mundo.
- Identificar las características positivas de cada una de las aplicaciones estudiadas para tenerlas en cuenta a la hora de desarrollar el sistema.
- Realizar un estudio sobre las posibles herramientas a utilizar.
- Caracterizar las herramientas de desarrollo a utilizar, teniendo en cuenta las ventajas y desventajas de las mismas, así como su influencia en el desarrollo de la aplicación.

- Realizar análisis y diseño del sistema.

Como se conoce, Cuba es un país que no posee un amplio desarrollo económico, y para estar a la altura de los países que si poseen una gran economía y dedican millones de dólares al desarrollo de este deporte, es necesario realizar diferentes inversiones en la automatización de las aplicaciones que se utilizan con estos fines. La solución propuesta reduce al país las inversiones en softwares de este tipo y en su lugar se podría invertir en otras necesidades del deporte.

DESARROLLO

Para el desarrollo del béisbol es importante que se lleven a cabo una serie de procesos, y que estos procesos se puedan efectuar con un alto nivel de calidad. Dentro de estos procesos se encuentra la anotación de los juegos. De la eficacia del proceso de anotación depende la veracidad de las estadísticas de este deporte. Para ello es importante automatizar la mayoría de estos procesos de forma tal que los mismos se ejecuten con rapidez y eficiencia.

Análisis del sistema actual

En Cuba se utiliza actualmente un sistema que lleva por nombre Béisbol, el cual permite optimizar un poco el proceso de anotación, que hasta el momento de su creación se desarrollaba sobre un papel y era un poco más engorroso. Ahora se cuenta con toda la información del partido jugada por jugada de manera digital, la cual se puede obtener en tres formatos diferentes: en .htm, para página web, en .doc, para imprimir sobre papel y en .list, para el trabajo con ficheros .bei, los cuales son compilados posteriormente.

Este sistema fue desarrollado por integrantes del Instituto de Educación Física y Recreación (INDER), en el año 1977 sobre MS-DOS ya que en Cuba la tecnología con que se contaba era muy pobre.

Sin embargo este sistema no tiene una interfaz de usuario amigable y resulta difícil el manejo del mismo para el anotador. Con el desarrollo tecnológico y la aparición de Windows con un sistema de ventanas muy fácil de usar que brinda una serie de facilidades para el manejo de un sistema como este, es necesaria la realización de una versión actualizada de la aplicación la cual hasta el momento no se ha realizado. Se hizo una actualización en 1994 pero fue para corregir algunos problemas y agregar algunas funcionalidades sobre la misma plataforma de MS-DOS.

Existe una copia de la aplicación en cada uno de los estadios del país. Durante el desarrollo de un juego se llevan todas las jugadas del encuentro a la misma y se envían mediante ficheros .bei al estadio Latinoamericano donde se procesa toda la información y se genera una página .htm que se muestra en un sitio web para que los aficionados puedan seguir el juego mientras se efectúa. Este proceso es lento y muchas veces no se tiene información del partido hasta que no se termina el inning en desarrollo o durante varios inning. Cuando se termina el juego se compilan todos los ficheros que se

obtuvieron en el estadio Latinoamericano y se envían de nuevo a cada uno de los estadios para actualizar la aplicación de forma independiente.

Muchos datos estadísticos se pierden, pues la información de cada lanzamiento no es registrada, debido a que la inserción de información estadística como se ha expresado en varias ocasiones solo ocurre cuando se completa una jugada.

Descripción General del Problema

Para llevar a cabo el proceso de anotación en los partidos de béisbol, en Cuba se utiliza un sistema de anotación y gestión de juegos de béisbol denominado “Béisbol”, concebido en 1977 y desarrollado en MS-DOS. Está provisto de una interfaz gráfica por consola dificultando así la interacción del anotador con el sistema, pero en el momento de su creación constituyó un proyecto innovador de gran ayuda para las tareas de gestión y anotación de los juegos de béisbol, que se realiza en papel y dificulta el trabajo de los anotadores y la gestión de las estadísticas que se generan a partir de la información recogida en cada partido. La anotación de los juegos como se explica anteriormente es en papel, el anotador debe tener un amplio conocimiento del Sistema Codificado de Anotación Cubano para desempeñar su tarea y paralelamente insertar en el sistema “Béisbol” cada jugada durante el encuentro, por lo que se realiza el mismo trabajo en dos ocasiones. Este sistema introdujo la posibilidad de guardar estadísticas en formato digital, sin embargo se pierden algunas estadísticas del juego pues los lanzamientos no se registran en el mismo.

Argumentación del problema a resolver

El sistema actual de anotación de juegos de béisbol en Cuba corre sobre MS-DOS, lo que trae consigo que se deriven una serie de dificultades con el uso del mismo por la interfaz que posee, además es necesario que el anotador conozca el sistema de codificación cubano, pues las jugadas se insertan en la aplicación mediante un código. Todos los datos del encuentro que se vayan a recoger están codificados, el nombre de los jugadores, los árbitros, anotadores oficiales, comisarios técnicos y las jugadas que pueden ocurrir en un partido.

Una vez comenzado el juego, cuando comparece al bate cada jugador, aparece una pantalla que recrea una serie de acciones posibles a la ofensiva y defensiva que

describen una jugada. A continuación se muestra el proceso por el que transcurre la anotación. Se muestra una pantalla que contiene 7 casillas:

En la casilla número 1 se pone el número del jugador que está consumiendo su turno al bate.

En la casilla número 2 la acción ofensiva del bateador.

En la casilla número 3 posiciones defensivas que intervienen en la acción ofensiva del bateador.

En la casilla número 4 las acciones defensivas ocurridas sobre la acción ofensiva del bateador.

En la casilla número 5 el avance del corredor a primera base.

En la casilla número 6 el avance del corredor a segunda base.

En la casilla número 7 el avance del corredor a tercera base.

Todas las acciones descritas anteriormente son insertadas en el sistema mediante un código equivalente a cada una de ellas, por lo que las combinaciones de números a tener en cuenta por el anotador para llevar a cabo la anotación de cada jugada en el partido pueden ser relativamente complejas y debe consultar el Sistema Codificado de Anotación o conocer a la perfección el mismo. Los lanzamientos que lleva a cabo cada pitcher en el encuentro no son tomados en cuenta por el sistema, lo que imposibilita almacenar una serie de estadísticas sobre los pitcher que intervienen en el partido.

La aplicación cuenta con grandes limitaciones tecnológicas y debido al desarrollo progresivo de los sistemas informáticos no se hace un buen uso de todas las potencialidades que brindaría un sistema de anotación de este tipo, pues de él se derivaría la posibilidad de contar con otra serie de servicios, como por ejemplo un sistema para mostrar las incidencias del juego en tiempo real, del cual el sistema de anotación sería la base para su funcionamiento.

Actualmente se llevan los juegos en vivo a través de un sitio web, pero por la complejidad del sistema y del funcionamiento del mismo, en ocasiones no se conoce lo que está sucediendo en tiempo real, pues los usuarios deben esperar a que transcurra un inning completo, para saber lo que sucede, y a veces hasta varios inning pues el sistema no envía los datos en tiempo real.

Metodologías

Existen varias metodologías de desarrollo que se aplican a la hora de desarrollar algunos sistemas y aplicaciones. A continuación se relacionan algunas características de la metodología escogida que permitieron la realización de la aplicación de una mejor forma.

Proceso Unificado de Software (RUP), como metodología utilizada en el desarrollo del trabajo.

El Proceso Unificado de Desarrollo de Software (RUP) constituye un proceso de desarrollo de software que junto al Lenguaje Unificado de Modelado (UML) conforman la Metodología más utilizada en la actualidad para el análisis, implementación y documentación de Sistemas Orientados a Objetos. Esta metodología puede ser adaptable al contexto de cada problema en particular, por lo tanto constituye una herramienta flexible, configurable y aplicable para la solución de numerosos sistemas.

Como RUP constituye un proceso, en su modelación define como sus principales elementos a los Trabajadores, Actividades, Artefactos y El Flujo de Actividades.

Trabajadores: especifica el comportamiento y responsabilidades de cada individuo, grupo de individuos, sistema automatizado o máquina que trabajan en conjunto.

Actividades: es una tarea que tiene un propósito definido, ejecutada por un trabajador y manipula elementos.

Artefactos: son el resultado de la culminación de las actividades, también pueden ser modificados y usados por estas. Pueden ser modelos, elementos de modelado, ejecutables y código fuente.

Flujo de Actividades: son una secuencia de actividades llevadas a cabo por los trabajadores y que produce un resultado de valor.

RUP esta basado en 5 principios de desarrollo, estos son:

Adaptar el proceso: el proceso debe adaptarse a las características particulares del proyecto, el tamaño del mismo, así como las regulaciones que condicionan su funcionamiento ya que estas influirán en su diseño específico.

Balancear propiedades: puesto que los requerimientos de los participantes en el proyecto pueden ser diferentes o contradictorios, se debe encontrar un balance que satisfaga de alguna manera los deseos de todos los integrantes.

Demostrar valor iterativamente: la entrega del proyecto, aunque sea de un modo interno, se realiza en etapas iteradas. En cada una de las iteraciones se analizan las opiniones de inversores, la estabilidad y calidad del producto para poder refinar la dirección del proyecto e identificar riesgos.

Elevar el nivel de abstracción: este principio dominante motiva el uso de conceptos reutilizables tales como el patrón de software. Esto evita que los ingenieros de software vayan directo de los requisitos a la codificación, sin saber con certeza qué codificar, para satisfacer de la mejor manera los requerimientos pensando en la reutilización de código.

Enfocarse en la calidad: el control de la calidad en el proyecto no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

El ciclo de vida de RUP

La vida de un sistema transcurre a través de ciclos de desarrollo, desde su nacimiento hasta su muerte, en cada ciclo se repite el proceso unificado de desarrollo, Cada ciclo consta de cuatro fases: Inicio, elaboración, construcción y transición. Cada ciclo concluye con una versión del producto.

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los

incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son mini proyectos.

Con la implantación de RUP como metodología para el desarrollo del trabajo en cuestión se obtiene una mayor organización en la concepción de elementos pues provee un marco de trabajo que facilita la aplicación de esta metodología a muchos de los problemas que se presentan en la vida real, ya que puede ser adaptable a las condiciones de cada problema en particular. [1]

Plataforma .NET

.NET es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Su objetivo es simplificar el desarrollo de aplicaciones Web. Provee las herramientas y tecnologías para transformar a Internet en una plataforma de computación distribuida en gran escala. Esta plataforma además soporta los estándares sobre los cuales se basan los servicios Web y utiliza tecnologías existentes, productos modificados para su uso dentro de la plataforma y elementos nuevos.

Plataforma .NET y Mono

.NET puede considerarse en ciertos aspectos como la respuesta de Microsoft a Java, aunque tiene bastantes diferencias. Reúne en una misma plataforma un conjunto interesante de características, como independencia de plataforma, independencia de lenguaje, soporte de bases de datos, soporte para XML, servicios Web y aplicaciones Web, entre otras. De entre todas estas, quizás una de las más importantes sean las dos primeras, referentes a la independencia tanto del lenguaje como de la plataforma.

Independencia del lenguaje

Se pueden desarrollar aplicaciones en múltiples lenguajes dentro de la plataforma .NET, pero lo más interesante, es que una aplicación puede tener diferentes partes desarrolladas en diferentes lenguajes, y todas estas pueden comunicarse entre sí, transparentemente, sin tener que utilizar ningún tipo de capa intermedia que posibilite esta comunicación. Esto permite a su vez una gran reutilización de código, ya que las clases

desarrolladas para un proyecto en un lenguaje concreto, podrán ser reutilizadas en un nuevo proyecto, independientemente del lenguaje en el que se desarrolle éste.

Independencia de plataforma

Al igual que en Java, el código .NET no se compila a código máquina, sino a un código en un formato intermedio, independiente de la plataforma. Esto permite llevar los binarios producidos de una plataforma a otra, tal como sucede en Java. En la plataforma .NET existe el Common Language Runtime (CLR), que se encarga de ejecutar el código intermedio o Common Intermediate Language (CIL).

Proyecto Mono

Mono es un proyecto de implementación del Framework .NET de Microsoft utilizando código libre (open source), gestionado por Ximian y basado en las especificaciones definidas en ECMA.

Mono es una implementación del Framework .NET, pero de nada más relacionado con .NET, como pasaporte o software como servicios.

Los objetivos iniciales del proyecto Mono eran implementar en un entorno de software libre para el mundo Unix la especificaciones ECMA, para lo cual se incluye un compilador para C#, un entorno de ejecución CLR y un conjunto de librerías de clase que incluyen las FCL (Framework Class Library), así como otras añadidas.

El compilador de Mono es el Mono Runtime, que sería el equivalente al Common Language Runtime o entorno virtual de ejecución en la plataforma .NET. Implementa un compilador en tiempo de ejecución Just In Time (JIT) para el CIL de la máquina virtual, un compilador antes del tiempo de ejecución Ahead-of-Time (AOT), un cargador de clases, un recolector de basura, el sistema de threadings y las librerías de acceso a los metadatos.

El entorno de ejecución se puede utilizar también embebido dentro de otra aplicación, de forma que se pueden extender aplicaciones C y C++ mediante C#.

En comparación con la solución ofrecida por Microsoft, Mono incluye un intérprete y la opción de utilizar el runtime de forma embebida en una aplicación, además funciona sobre diversos sistemas operativos:

Linux, MacOS X, BSD, SUN SOLARIS y Microsoft Windows. Aparte el intérprete puede ejecutarse sobre otras plataformas como HP-UX.

Mono constituye una opción para la portabilidad de aplicaciones desarrolladas en C #, puesto que brinda la posibilidad de hacer funcionar estas aplicaciones en una amplia variedad de sistemas operativos incrementando el alcance de las mismas.

Mono contiene una serie de elementos útiles para la construcción de un software:

- Un lenguaje común de infraestructura (CLI) máquina virtual que contiene un cargador de clases, justo a tiempo de compilación, recolección de basura y un tiempo de ejecución.
- Una biblioteca de clases que puede trabajar con cualquier lenguaje que trabaja en el CLR. Mono y .NET compatibles con las bibliotecas de clases.
- Un compilador para el lenguaje C #.

El CLR y el Sistema de Tipo Común (CTS) permiten que las aplicaciones y bibliotecas que se escriben en una colección de diferentes lenguajes se centren en el código de bytes. Esto significa por ejemplo que si se define una clase para hacer la manipulación algebraica en C #, esa clase puede ser reutilizada por cualquier otro lenguaje que apoye el CLI. Usted podría crear una clase en C #, subclase en C + + y crear una instancia en un programa Eiffel. Un solo sistema objeto, sistema de hilos, bibliotecas de clases, y sistema de recolección de basura, pueden ser compartidos a través de todos estos lenguajes. [2]

XML

XML (eXtensible Markup Language) es un lenguaje orientado a identificar estructuras de datos en un documento. La especificación XML define la manera estándar de cómo hay que realizar el marcado de expresiones en un documento no estructurado, para que con dicho marcado se defina una determinada estructura de datos.

La especificación XML no define el contenido de las estructuras de datos, son los expertos de cada dominio y las entidades reguladoras, los agentes que pueden utilizar el estándar XML para consensuar un lenguaje común que permita transformar los documentos no estructurados en estructuras procesables por un sistema.

Cuando hablamos de un documento nos referimos no solo al concepto tradicional de documento en papel o soporte electrónico sino a todos los tipos de documentos actuales: páginas Web, correo electrónico, gráficos vectoriales, transacciones de comercio electrónico, etc. Un documento XML es un documento que puede ser leído y entendido

por una persona "human readable system" y a la vez puede ser procesado por un sistema para extraer información "machine readable system". [3]

Ventajas

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML. Esto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan bugs y se acelera el desarrollo de aplicaciones.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones.

Herramientas utilizadas

Para el desarrollo de un software es necesario utilizar algunas herramientas que permiten la creación de este y que suelen hacer el trabajo más fácil.

Visual Studio 2005

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE) que provee un conjunto de herramientas de desarrollo diseñadas para ayudar a los desarrolladores de software (tanto si son principiantes como profesionales con experiencia) a enfrentarse a los desafíos complejos y crear soluciones innovadoras. Cada día, los desarrolladores de software se abren camino a través de duros problemas para crear software que suponga una diferencia en la vida de los demás. La función de Visual Studio es mejorar el proceso de desarrollo para hacer que el trabajo de lograr esos adelantos sea más fácil y satisfactorio.

Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Así se

pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles. [4]

Los desarrolladores de este trabajo escogieron esta herramienta porque:

- Aumenta la productividad y la calidad gracias a herramientas integradas y conocidas.
- Implementa y protege la infraestructura y las aplicaciones web críticas, así como ofrece compatibilidad entre ellas.
- Reduce los costos mediante una mejor visibilidad del proceso de desarrollo.
- Ofrece mejor capacidad de predicción y planeación mediante la compatibilidad integrada con procesos y metodologías.
- Tienen un conocimiento previo de la misma y tienen experiencia en el manejo de esta.

Lenguaje de programación

C Sharp (C#)

Microsoft Visual C# 2005 es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente y mantiene la expresividad y elegancia de los lenguajes de tipo C. [5]

Visual Studio admite Visual C# con un editor de código completo, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y fácil de usar, además de otras herramientas. La biblioteca de clases .NET Framework ofrece acceso a una amplia gama de servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa.

En la actualidad existen los siguientes compiladores para el lenguaje C#:

- Microsoft.NET framework SDK incluye un compilador de C#, pero no un IDE.
- Microsoft Visual Studio .NET, IDE por excelencia de este lenguaje, versión 2002, 2003, 2005 y 2008.
- #develop, es un IDE libre para C# bajo licencia LGPL, muy similar a Microsoft Visual C#.

- Mono, es una implementación GPL de todo el entorno .NET desarrollado por Novell. Como parte de esta implementación se incluye un compilador de C#.
- Delphi 2006, de Borland Software Corporation.
- dotGNU Portable.NET, de la Free Software Foundation

Se escoge como lenguaje de programación por las ventajas que ofrece, las cuales se enuncian a continuación:

- Declaraciones en el espacio de nombres: al empezar a programar algo, se puede definir una o más clases dentro de un mismo espacio de nombres.
- Tipos de datos: en C# existe un rango más amplio y definido de tipos de datos que los que se encuentran en C, C++ o Java.
- Atributos: cada miembro de una clase tiene un atributo de acceso, que puede ser de tipo público, protegido, interno, interno protegido y privado.
- Pase de parámetros: aquí se puede declarar a los métodos para que acepten un número variable de parámetros. De forma predeterminada, el pase de parámetros es por valor, a menos que se use la palabra reservada ref, la cual indica que el pase es por referencia.
- Métodos virtuales y redefiniciones: antes de que un método pueda ser redefinido en una clase base, debe declararse como virtual. El método redefinido en la subclase debe ser declarado con la palabra override.
- Propiedades: un objeto tiene intrínsecamente propiedades, y debido a que las clases en C# pueden ser utilizadas como objetos, C# permite la declaración de propiedades dentro de cualquier clase.
- Inicializador: un inicializador es como una propiedad, con la diferencia de que en lugar de un nombre de propiedad, un valor de índice entre corchetes se utiliza en forma anónima para hacer referencia al miembro de una clase.
- Control de versiones: C# permite mantener múltiples versiones de clases en forma binaria, colocándolas en diferentes espacios de nombres. Esto permite que versiones nuevas y anteriores de software puedan ejecutarse en forma simultánea.

Aunque tiene muchas ventajas también tiene sus desventajas, entre ellas se encuentran:

- Se tiene que conseguir una versión reciente de Visual Studio .NET.
- Se debe contar con algunos requerimientos mínimos del sistema para poder trabajar adecuadamente tales como: contar con Windows NT 4 o superior, tener alrededor de 4 gigas de espacio libre para la pura instalación, etc.
- Además para quien no está familiarizado con ningún lenguaje de programación, le costará más trabajo iniciarse en su uso, y si se quiere consultar algún tutorial más explícito sobre la programación en C# se tendría que contar además con una conexión a Internet. [6]

Problema y situación problemática

La Federación Cubana de Béisbol Amateur lleva a cabo una serie de procesos vinculados al béisbol. La anotación de estos juegos es uno de los procesos mas importantes que se desarrollan, pues es la base de la recopilación de estadísticas que son un aspecto fundamental para este deporte. La aplicación actual que se utiliza para realizar la anotación no cumple con las necesidades actuales del béisbol en Cuba, lo que forma parte de la situación problemática del trabajo

El sistema actual que se encarga de la anotación de juegos de béisbol está desarrollado en MS-DOS, porque en el momento en que se realizó, en Cuba no existía un desarrollo tecnológico como el que existe en la actualidad. Es necesario mejorar la interfaz que posee esta aplicación para permitir que los usuarios del mismo se sientan atraídos y le sea más fácil cumplir con su trabajo.

Es importante también permitir que los juegos se anoten lanzamiento por lanzamiento, lo que lograría recopilar mayor cantidad de información estadística acerca de los lanzadores del encuentro, algo que actualmente no se hace, pues solamente se insertan datos en la aplicación cuando se completa una jugada. Además es necesario que el anotador pueda hacer su trabajo de una forma más rápida y eficiente, esto se lograría eliminando la forma en que los anotadores insertan una jugada en el sistema.

Actualmente lo hacen mediante un código, lo que provoca que su trabajo sea más engorroso y los obliga a conocer profundamente el Sistema Codificado de Anotación Cubano.

Propuesta del sistema

Para dar solución a la problemática planteada será implementado un sistema de anotación que contará con un conjunto de características que optimizarán las funcionalidades de la aplicación existente. Dentro de las nuevas mejoras se encuentra la implementación de una interfaz de usuario amigable, facilitando la interacción del usuario con la aplicación, ya que el sistema anterior estaba desarrollado en MS-DOS y restringía de forma considerable el uso de la misma. También se incorpora la funcionalidad de anotar lanzamiento por lanzamiento y no jugada por jugada como se anotaba anteriormente; esta nueva funcionalidad permite obtener un conjunto de estadísticas, como: conocer la cantidad de lanzamientos que efectúa un pitcher, los tipos de lanzamientos que realiza, la velocidad y la zona donde cayó cada lanzamiento realizado durante el encuentro. La anotación de las jugadas ocurridas durante el encuentro será mostrada al anotador de forma descriptiva, el anotador escoge la jugada correcta y el sistema se encarga de codificar y guardar la información, por lo que se logra una mayor eficiencia y veracidad en la recopilación de los datos estadísticos.

El sistema debe ser desplegado en cada estadio del país con el propósito de anotar todos los partidos que se lleven a cabo en cada jornada de juegos. Debe contar con una conexión de red para poder acceder al servidor central mediante un Servicio Web, incorpora así la funcionalidad de salvar todos los datos estadísticos recopilados durante el juego en un fichero .XML, que posteriormente será enviado al servidor.

El proceso de envío del fichero es realizado automáticamente por la aplicación mientras se desarrolle el partido, actualizando la información que se encuentra en el servidor. En caso de no haber conexión en un momento dado, los datos se seguirán guardando en el fichero .XML y se continuará haciendo peticiones de conexión al servidor, y en cuanto se restablezca la conexión se envía el fichero y se actualiza la información en el servidor. Esta nueva funcionalidad abre paso a la posibilidad de crear una aplicación Web de seguimiento de juego en tiempo real mejorando así las prestaciones del nuevo sistema de anotación.

CONCLUSIONES

Con el desarrollo del trabajo en cuestión se han cumplido los objetivos planteados en el mismo. Como resultado, se ha obtenido el Sistema de Anotación de Juegos de Béisbol que ayudará de manera ostensible la labor de anotación que lleva a cabo la FCBA en todos los juegos desarrollados en la Serie Nacional y otros eventos que tienen lugar en Cuba. Para la concepción de la aplicación fue necesario realizar todo un Proceso de Desarrollo de Software que condujo a la obtención de un producto que cumple las expectativas de la FCBA.

El uso del sistema en cuestión permite a los anotadores desarrollar su trabajo de una forma más amena y eficiente, sin la necesidad de conocer el código de anotación profundamente. Provee la nueva posibilidad de anotar lanzamiento por lanzamiento y las jugadas se insertan mediante una descripción lógica de los sucesos ocurridos durante el desarrollo de las mismas. La codificación es realizada automáticamente por la propia aplicación, minimizando la posibilidad de la existencia de errores a la hora de anotar las jugadas.

El sistema cuenta con las siguientes facilidades:

- Muestra las estadísticas de los jugadores que participan o participaron durante el juego.
- Permite anotar lanzamiento por lanzamiento.
- El usuario debe autenticarse antes de usar el sistema por lo que impide que personal ajeno a la FCBA pueda acceder a la misma.
- Lleva el conteo de los lanzamientos realizados por cada pitcher que participa en el juego.
- Muestra un marcador en tiempo real que visualiza el resultado parcial del encuentro.

Todas las ventajas anteriormente descritas hacen del sistema implementado una solución factible que favorece al proceso de anotación de juegos de béisbol, constituyendo una herramienta de trabajo para los anotadores, la cual facilita su desempeño profesional y es el reflejo de una nueva generación de aplicaciones que fomentan el desarrollo del béisbol en Cuba.

REFERENCIAS BIBLIOGRÁFICAS

1. Innova, Grupo Soluciones. GSInnova. 2007; <http://www.rational.com.ar>.
2. Echarte, Francisco. eslomas.com. EsLoMas. 2008; <http://www.eslomas.com>.
3. vico open modeling, s.l. Vico.org: Open Modeling. Vico.org. 2008; <http://www.vico.org>.
4. Corporation, Microsoft. msdn. msdn. 2008; <http://msdn.microsoft.com>.
5. Corporation, Microsoft. msdn. msdn. 2009; <http://msdn.microsoft.com>.
6. LWP. LWP. 2009; <http://www.lawebdelprogramador.com>.

BIBLIOGRAFÍA

Pérez Julien, José y Mendez Rodríguez, Efrain. Reglas Oficiales de Béisbol 2008. Ciudad de la Habana: s.n., 2008.

González Valnuerdi, Eduardo. Sistema Codificado de Anotación Béisbol. Manual. Ciudad de la Habana: CINID-INDER, 1992.

Merelo, J. J. Geneura. [En línea] <http://geneura.ugr.es/~jmerelo/tutoriales/csharp/>.

Echarte, Francisco. eslomas.com. EsLoMas. [En línea] 2008. <http://www.eslomas.com>.

W3C. World Wide Web Consortium. [En línea] <http://www.w3c.es/divulgacion/guiasbreves/tecnologiasXML>.

LWP. La Web del Programador. [En línea] <http://www.lawebdelprogramador.com/>.

djk. Devjoker. [En línea] [http://www.devjoker.com/contenidos/Articulos/280/Desencriptar-Texto-y-variables-con-NET-VB-y-C-\(segunda-parte\).aspx](http://www.devjoker.com/contenidos/Articulos/280/Desencriptar-Texto-y-variables-con-NET-VB-y-C-(segunda-parte).aspx).

ANEXOS

Imágenes del sistema implementado



Figura 1: Pantalla de inicio del sistema.

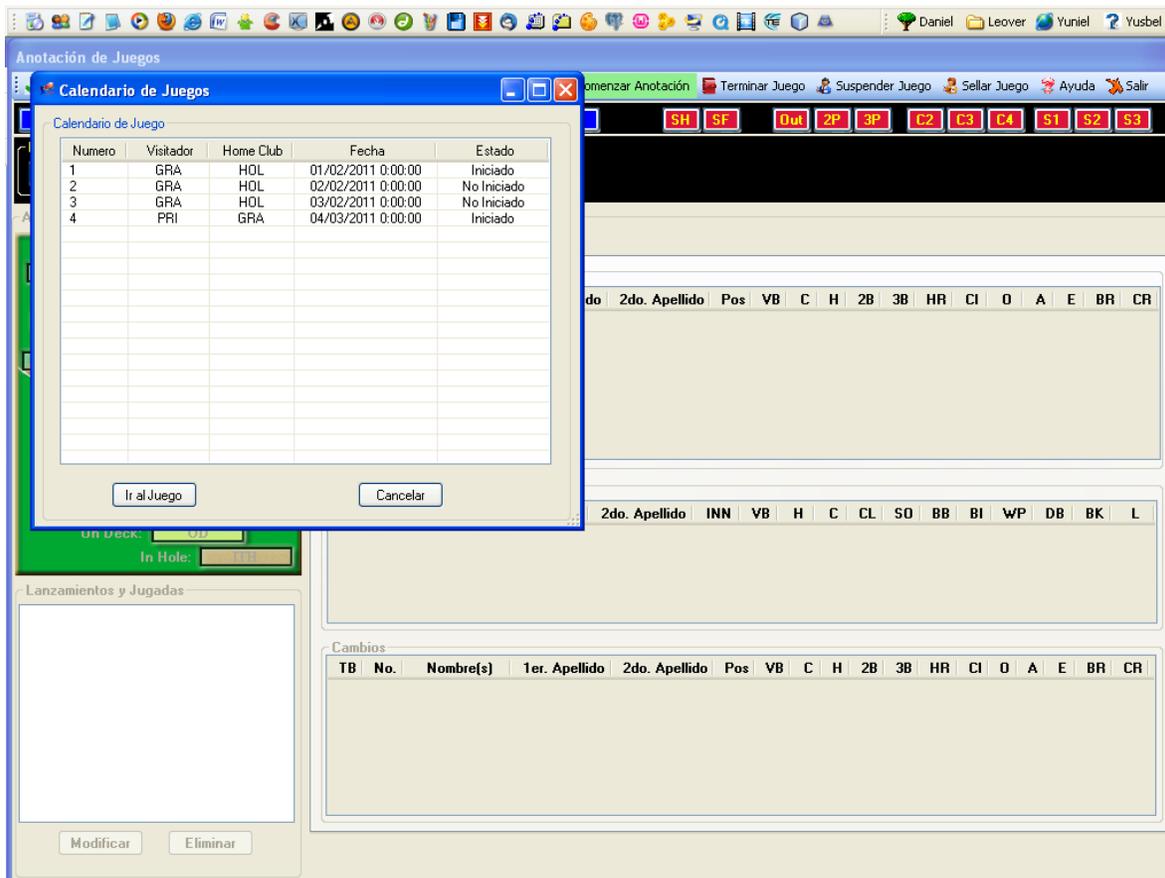


Figura 2: Escoger un juego del calendario para anotar.

Anotación de Juegos

Seleccionar Juego Rosters Alineación VS Alineación HC Datos del Juego Comenzar Anotación Terminar Juego Suspender Juego Sellar Juego Ayuda Salir

H 2B 3B HR DB BBI BK WP PB R2 R3 R4 E SH SF Out 2P 3P C2 C3 C4 S1 S2 S3

Lanzamientos: B BR SC ST F Inning: 3 Bola(s): 0 Equipos: 1 2 3 4 5 6 7 8 9 C H E
 Al Bate: GRA Strike(s): 0 GRA 0 9 36 45 48 0
 Out(s): 1 HOL 10 6 16 19 0

Anotación del Juego

Visitor Home Club

Alineación

TB	No.	Nombre(s)	1er. Apellido	2do. Apellido	Pos	VB	C	H	2B	3B	HR	CI	O	A	E	BR	CR
1	20	Ramón	Tamayo	Tamayo	2B	6	5	6	0	0	2	7	1	2	0	0	0
2	14	Urmanis	Guerra	Vargas	RF	6	5	6	2	0	0	1	0	0	0	0	0
3	51	Yoenis	Céspedes	Milanés	CF	6	5	5	0	1	2	9	2	0	0	0	0
4	54	Alfredo	Despaigne	Rodríguez	LF	5	5	4	0	0	3	7	1	0	0	0	0
5	8	Yordanis	Samón	Matamoros	1B	6	5	5	0	0	1	2	2	0	0	0	0
6	31	Carlos	Barrabí	García	R	6	5	5	0	0	1	4	0	0	0	0	0
7	5	Carlos	Benítez	Pérez	3B	6	5	5	0	0	1	0	0	0	0	0	0
8	60	Luis Alberto	Ferrales	Jiménez	BD	6	5	6	2	0	0	5	0	0	0	0	0
9	7	Adrián	Moreno	Rondón	SS	6	5	6	0	1	1	9	0	1	0	0	0

Pitchers

No.	Nombre(s)	1er. Apellido	2do. Apellido	INN	VB	H	C	CL	SO	BB	BI	WP	DB	BK	L
62	Ciro Silvino	Licea	González	2	24	19	16	16	0	0	0	0	0	0	24

Cambios

TB	No.	Nombre(s)	1er. Apellido	2do. Apellido	Pos	VB	C	H	2B	3B	HR	CI	O	A	E	BR	CR
----	-----	-----------	---------------	---------------	-----	----	---	---	----	----	----	----	---	---	---	----	----

On Deck: Guerra
In Hole: Céspedes

Lanzamientos y Jugadas

- Urmanis Guerra
- Yoenis Céspedes
- Alfredo Despaigne
- Yordanis Samón
- Carlos Barrabí
- Carlos Benítez
- Luis Alberto Ferrales
- Adrián Moreno

En Jugada - Otro: 0 milla
Hit al Jardín Central de F

Modificar Eliminar

Figura 3: Pantalla de anotación general.

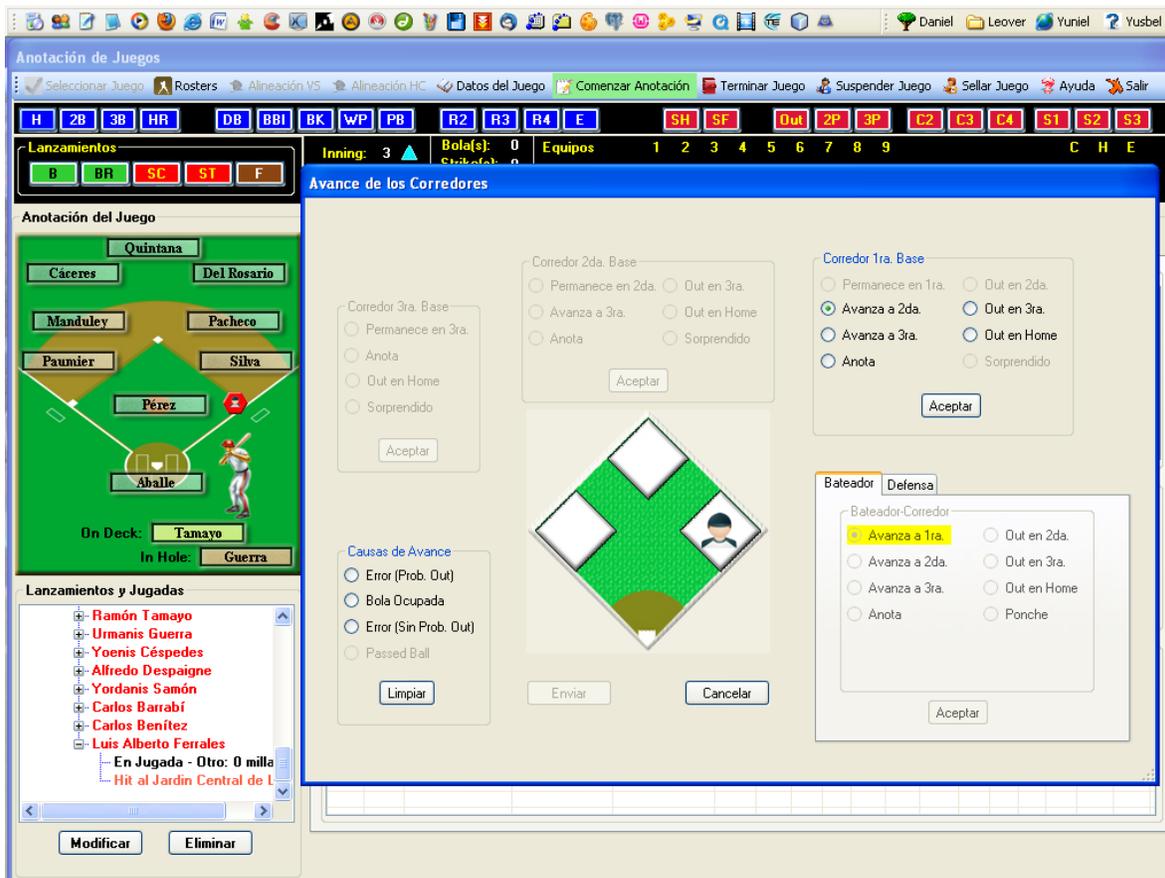


Figura 4: Pantalla de anotación ofensiva.

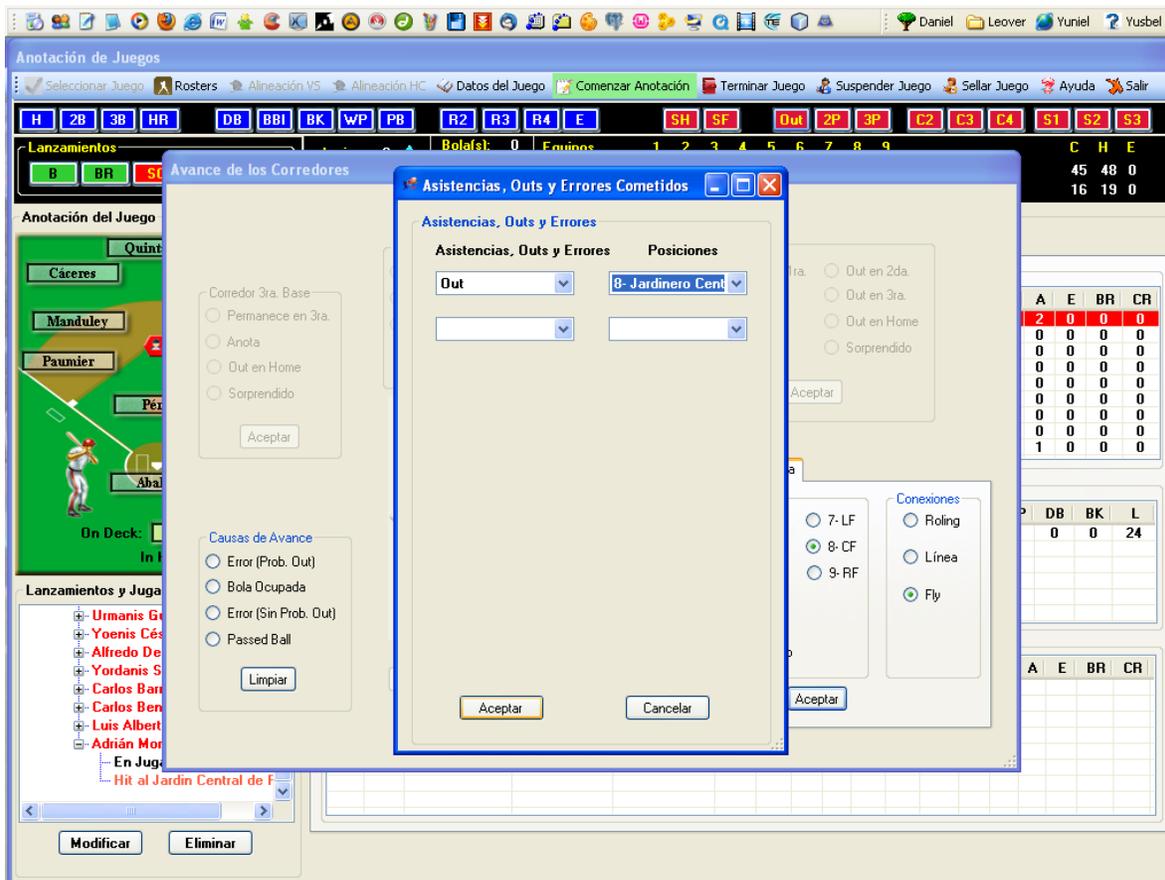


Figura 5: Pantalla de anotación defensiva.

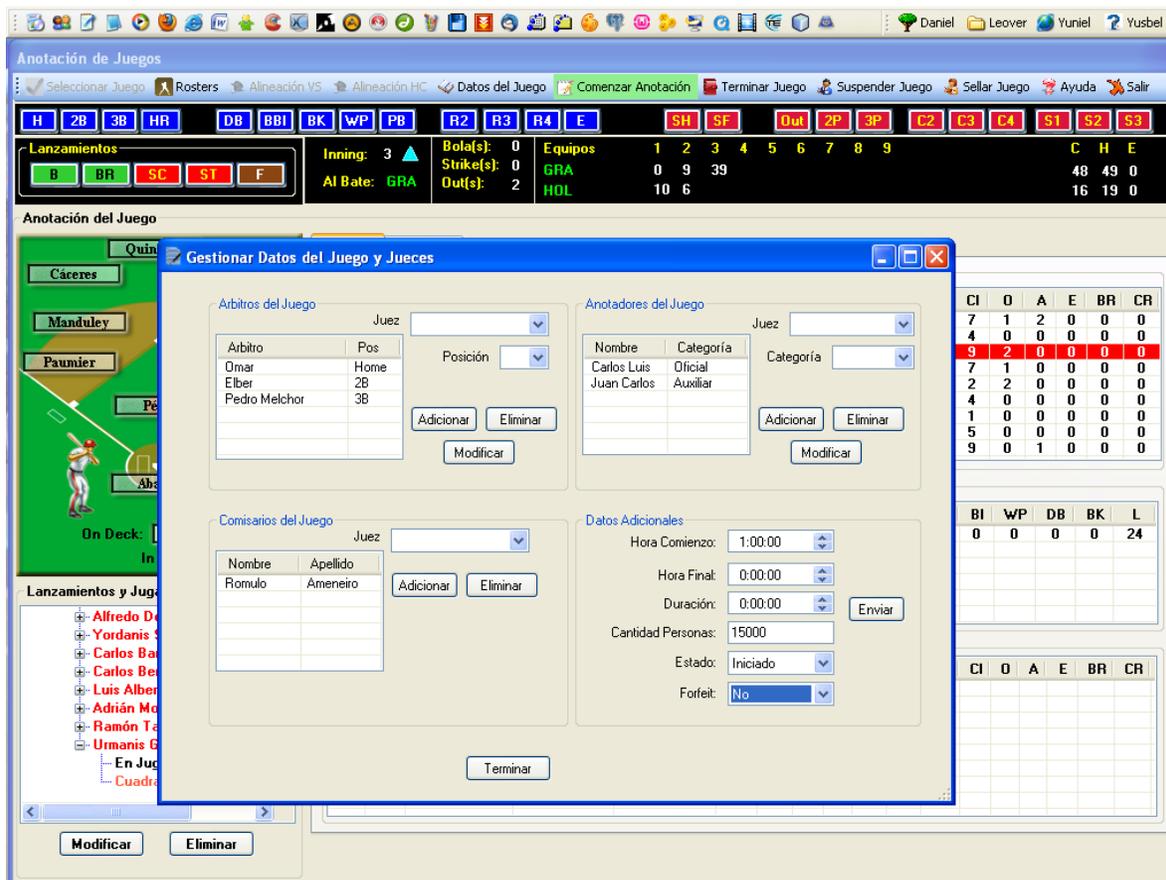


Figura 6: Anotar los datos generales del juego.