



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE OLIMPIADAS
INTERUNIVERSITARIAS EN EL ÁREA DE TECNOLOGÍA DE
LA ESCUELA DE CIENCIAS Y SISTEMAS, USAC**

Jorge Luis Arroyo Pérez

Asesorado por el Ing. Gandy Ricardo Tejeda Castañaza

Guatemala, octubre de 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE OLIMPIADAS
INTERUNIVERSITARIAS EN EL ÁREA DE TECNOLOGÍA DE
LA ESCUELA DE CIENCIAS Y SISTEMAS, USAC**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

JORGE LUIS ARROYO PÉREZ

ASESORADO POR EL ING. GANDY RICARDO TEJEDA CASTAÑAZA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, OCTUBRE DE 2014

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympto Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympto Paiz Recinos
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
EXAMINADOR	Ing. Luis Fernando Quiñones López
EXAMINADOR	Ing. José Alfredo Gonzáles Díaz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE OLIMPIADAS INTERUNIVERSITARIAS EN EL ÁREA DE TECNOLOGÍA DE LA ESCUELA DE CIENCIAS Y SISTEMAS, USAC

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 23 de octubre de 2013.



Jorge Luis Arroyo Perez

Guatemala, 14 de enero de 2014

Ing. Silvio José Rodríguez Serrano
Director de la unidad de EPS
Facultad de ingeniería
Presente

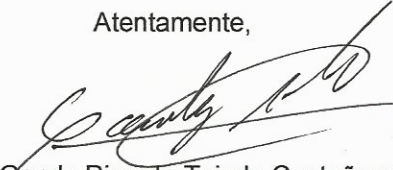
Estimado Ingeniero Rodríguez Serrano.

Por este medio atentamente le informa que como Asesor-Supervisor de la Práctica del Ejercicio Profesional Supervisado (E.P.S.), del estudiante universitario **Jorge Luis Arroyo Pérez** de la Carrera de Ingeniería en Ciencias y Sistemas, con carné No. **200511658**, procedí a revisar el informe final, cuyo título es "**SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE OLIMPIADAS INTERUNIVERSITARIAS EN EL ÁREA DE TECNOLOGÍA DE LA ESCUELA DE CIENCIAS Y SISTEMAS, USAC**".

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,


Ing. Gandy Ricardo Tejada Castañaza
Asesor de EPS
Colegiado No. 7618

Gandy Ricardo Tejada Castañaza
Km. 26.5 Carretera a El Salvador
Lote 23 Manzana 8 Lotificación Bosques de
Villa Verde (Villa Canales)
Teléfono: 57412444
No. Cel: 99999999



Guatemala, 14 de mayo de 2014.
REF.EPS.DOC.567.05.2014.

Ing. Silvio José Rodríguez Serrano
Director Unidad de EPS
Facultad de Ingeniería
Presente

Estimado Ingeniero Rodríguez Serrano .

Por este medio atentamente le informo que como Supervisora de la Práctica del Ejercicio Profesional Supervisado, (E.P.S) del estudiante universitario de la Carrera de Ingeniería en Ciencias y Sistemas, **Jorge Luis Arroyo Pérez** carné No. **200511658** procedí a revisar el informe final, cuyo título es **SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE OLIMPIADAS INTERUNIVERSITARIAS EN EL ÁREA DE TECNOLOGÍA DE LA ESCUELA DE CIENCIAS Y SISTEMAS.**

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

"Id y Enseñad a Todos"

Inga. Floriza Felipa Avila Pesquera de Medinilla
Supervisora de EPS
Área de Ingeniería en Ciencias y Sistemas

FFAPdM/RA





Guatemala, 14 de mayo de 2014.
REF.EPS.D.258.05.2014.

Ing. Marlon Antonio Pérez Turk
Director Escuela de Ingeniería Ciencias y Sistemas
Facultad de Ingeniería
Presente

Estimado Ingeniero Perez Turk.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE OLIMPIADAS INTERUNIVERSITARIAS EN EL ÁREA DE TECNOLOGÍA DE LA ESCUELA DE CIENCIAS Y SISTEMAS**, que fue desarrollado por el estudiante universitario **Jorge Luis Arroyo Pérez carné No. 200511658** quien fue debidamente asesorado por el Ing. Gandy Ricardo Tejeda Castañaza y supervisado por la Inga. Floriza Felipa Ávila Pesquera de Medinilla.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor y la Supervisora de EPS, en mi calidad de Director apruebo su contenido solicitándole darle el trámite respectivo.

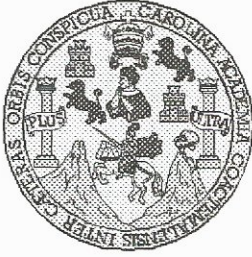
Sin otro particular, me es grato suscribirme.

Atentamente,
"Id y Enseñad a Todos"

Ing. Silvio José Rodríguez Serrano
Director Unidad de EPS

SJRS/ra





Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 10 de Septiembre de 2014

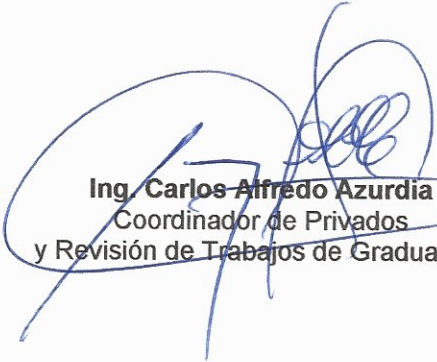
Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

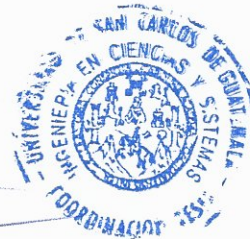
Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación-EPS del estudiante **JORGE LUIS ARROYO PÉREZ**, carné **2005-11658**, titulado: **"SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE OLIMPIADAS INTERUNIVERSITARIAS EN EL ÁREA DE TECNOLOGÍA DE LA ESCUELA DE CIENCIAS Y SISTEMAS, USAC"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



E
S
C
U
L
A

D
E

C
I
E
N
C
I
A
S

Y

S
I
S
T
E
M
A
S

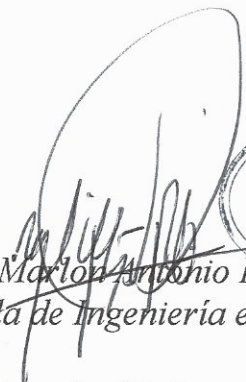

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación **“SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE OLIMPIADAS INTERUNIVERSITARIAS EN EL ÁREA DE TECNOLOGÍA DE LA ESCUELA DE CIENCIAS Y SISTEMAS, USAC”**, realizado por el estudiante JORGE LUIS ARROYO PÉREZ, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”



Ing. Maxlón Antonio Pérez Türk
Director, Escuela de Ingeniería en Ciencias y Sistemas
Guatemala, 27 de octubre 2014



El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE OLIMPIADAS INTERUNIVERSITARIAS EN EL ÁREA DE TECNOLOGÍA DE LA ESCUELA DE CIENCIAS Y SISTEMAS, USAC**, presentado por el estudiante universitario: **Jorge Luis Arroyo Pérez**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.


Ing. Murphy Olympo Paiz Recinos
Decano



Guatemala, octubre de 2014

/cc

AGRADECIMIENTOS A:

Ing. Gandy Tejeda

Por su valioso tiempo y ayuda en la realización de este trabajo, por su amable disposición y asesoramiento en el transcurso de este proyecto.

La Escuela de Sistemas

Por haberme dado la oportunidad de realizar este proyecto.

Ing. Francisco Pineda

Por su ayuda abnegada y desinteresada, su apoyo incondicional hasta el momento, por haberme escuchado y asesorado.

Mis amigos

Rodrigo López, Alexander Lemus, Alexander Martínez y a todos y cada uno de aquellos que siempre están, estuvieron y seguirán estando, brindándome su estima y soporte.

A todos

Y cada uno de aquellos que de alguna forma, colaboraron en la realización de este trabajo, hago extensivo mi más sincero agradecimiento.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES	III
LISTA DE SÍMBOLOS	V
GLOSARIO	VII
RESUMEN	XIII
OBJETIVOS	XV
INTRODUCCIÓN	XVII
1. FASE DE INVESTIGACIÓN	1
1.1. Antecedentes de la empresa	1
1.1.1. Reseña histórica	1
1.1.2. Misión	3
1.1.3. Visión	3
1.1.4. Servicios que realiza	4
1.2. Descripción de las necesidades	4
1.3. Priorización de las necesidades	6
2. FASE TÉCNICO PROFESIONAL	7
2.1. Descripción del proyecto	7
2.1.1. Preparación del evento	7
2.1.1.1. Problemas de programación	7
2.1.1.2. Ponderación de problemas	8
2.1.1.3. Inscripción de grupos	9
2.1.1.4. Protocolo del evento	9
2.1.2. Herramienta de administración	10
2.2. Investigación preliminar para la solución del proyecto	11

2.2.1.	Concurso de programación.....	11
2.2.2.	Software de administración de eventos	13
2.3.	Presentación de la solución al proyecto	14
2.3.1.	Metodología de trabajo (Scrum)	14
2.3.2.	Plan de trabajo.....	16
2.4.	Costos del proyecto.....	17
2.4.1.	Presupuesto.....	17
2.4.2.	Recursos.....	21
2.5.	Beneficios del proyecto	21
3.	FASE ENSEÑANZA APRENDIZAJE	23
3.1.	Capacitación propuesta.....	23
3.1.1.	Capacitación del protocolo del evento	23
3.1.2.	Capacitación de la herramienta	23
3.2.	Material elaborado.....	23
	CONCLUSIONES.....	25
	RECOMENDACIONES	27
	BIBLIOGRAFÍA.....	29
	APÉNDICES.....	31

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Cronograma de actividades	16
2.	Especificaciones del sitio	18

TABLAS

I.	Necesidades	5
II.	Priorización de necesidades	6
III.	Ponderación de problemas	8
IV.	Programming Contest	12
V.	Programming Contest Software	13
VI.	Especificaciones de hardware.....	17
VII.	Costo del sitio.....	17

LISTA DE SÍMBOLOS

Símbolo	Significado
&	Amberson
@	Arroba
\$	Dólar estadounidense
%	Porcentaje
+	Signo de más

GLOSARIO

.NET Compact Framework

Entorno independiente del hardware para ejecutar programas en dispositivos informáticos con recursos insuficientes. Hereda la arquitectura completa de .NET Framework de Common Language Runtime, es compatible con un subconjunto de la biblioteca de clases de .NET Framework y contiene clases diseñadas exclusivamente para .NET Compact Framework.

.NET Framework

Componente integral de Windows que admite la creación, implementación y ejecución de la siguiente compilación de aplicaciones y servicios web. Proporciona un entorno de múltiples lenguajes basado en estándares y muy productivo para integrar las inversiones existentes con aplicaciones y servicios de la próxima generación, así como la agilidad necesaria para resolver los desafíos que suponen la implementación y el funcionamiento de las aplicaciones para Internet.

ADO.NET

Conjunto de tecnologías de acceso a datos incluidas en las bibliotecas de clases de .NET Framework que proporcionan acceso a datos relacionales y a XML. ADO.NET está formado de clases que conforman el

conjunto de datos, proveedores de datos de .NET Framework y definiciones de tipos.

Adaptador del *host* En el modelo de programación de complementos de .NET Framework, es un ensamblado que contiene una o varias clases y convierte los datos entre la vista de host del complemento y el contrato. En función de la dirección de la llamada, el adaptador convertirá una vista en un contrato o un contrato en una vista.

Almacén de datos Conjunto de archivos XML del equipo de escritorio que contiene información sobre las plataformas, dispositivos, emuladores y paquetes que se instalan en el equipo.

Aplicación *host* Aplicación que se tiene alguna personalización, como un complemento de nivel de aplicación o una personalización de nivel de documento para la que la aplicación actúa como host.

Archivo de configuración Archivo XML con la extensión .config que contiene la configuración de las opciones para una aplicación o sitio Web. Los archivos de configuración comunes incluyen Machine.config y Web.config.

Autopostback	En controles de servidor ASP.NET, valor de configuración que hace que el control envíe la página. Cuando el usuario interactúa con el control. (De forma predeterminada, solo los controles de botón producen una devolución de datos).
C#	Lenguaje de programación diseñado para crear aplicaciones empresariales que se ejecutan en .NET Framework. C#, que es una evolución de C y C++, garantiza la seguridad de tipos y está orientado a objetos.
Caché de ensamblado	Caché de código para todo el equipo que almacena los ensamblados instalados específicamente para ser compartidos por varias aplicaciones del equipo. Estos deben tener nombres seguros.
Common Language Runtime	Moto que es el núcleo de la ejecución de código administrado. El motor en tiempo de ejecución proporciona al código administrado servicios como integración entre varios lenguajes, seguridad de acceso a código, administración de la duración de los objetos y compatibilidad con la depuración y la generación de perfiles.

Control de servidor HTML	Control de servidor ASP.NET que pertenece al espacio de nombres System.Web.UI.HtmlControls. Un control de servidor HTML se asigna directamente a un elemento HTML y se declara en una página ASP.NET como un elemento HTML marcado por un atributo runat="server",
Control de usuario	En ASP.NET: control de servidor creado de manera declarativa utilizando la misma sintaxis que una página ASP.NET y guardado como un archivo de texto con la extensión .ascx.
Control dinámico	Control de elementos Web que se conserva en un almacén de personalización; no aparece en el marcado declarativo de una página .aspx. Cuando se agrega a una página, el control WebPartManager crea automáticamente una instancia del control a partir del almacén de personalización en futuras solicitudes.
Control estático	Control de elementos Web que forma parte del marcado de persistencia de página declarativa en una página .aspx. Solo existe como parte de la página .aspx, no en un almacén de personalización como un control dinámico de elementos Web. A diferencia de un control dinámico, un control estático se agrega a una página .aspx en cada solicitud.

Control <i>host</i>	Objeto nativo de una aplicación host (como Microsoft Office Word o Microsoft Office Excel) al que se agregan enlaces de datos y eventos. Algunos ejemplos de controles host son los controles Bookmark y los controles ListObject.
Datos dinámicos	En ASP.NET marco que facilita la creación de aplicaciones web controladas por datos. Los datos dinámicos utilizan páginas personalizables y pantallas de campo.
Ensamblado	Conjunto de uno o varios archivos que pertenecen a una versión y se implementan como unidad. Un ensamblado es el bloque de creación principal de una aplicación .NET Framework. Todos los tipos y recursos administrados se incluyen en un ensamblado y se marcan como accesibles únicamente dentro del ensamblado o bien como accesibles desde código de otros ensamblados.
Estado de sesión	En ASP.NET, almacén de variables creado en el servidor para el usuario actual; cada usuario mantiene un estado de sesión independiente en el servidor. El estado de sesión se utiliza normalmente para almacenar información específica del usuario entre las devoluciones de datos.

Esquemas XML (XSD)

Lenguaje estándar de World Wide Web Consortium (W3C) utilizado para crear documentos de esquema XML.

Host

En el modelo de programación de complementos de .NET Framework, es el ensamblado de la aplicación host que se comunica con un complemento a través de la canalización de comunicación.

Metadatos

Información que describe todos los elementos administrados por Common Language Runtime: un ensamblado, el archivo cargable, el tipo, el método, etc. Esto puede incluir información necesaria para la depuración y la recolección de elementos no utilizados, así como atributos de seguridad, cálculo de referencias de datos, definiciones extendidas de clases y miembros.

Modo de presentación

Distintos estados de presentación que se pueden introducir en una página de elementos Web, que permiten a los usuarios modificar una página de manera especificada.

Web Forms

Marco de trabajo de paginas ASP.NET, compuesto por páginas Web programables (denominadas páginas de formularios Web Forms) que contienen controles de servidor reutilizables.

RESUMEN

Desde ya hace varios años en la Universidad de San Carlos de Guatemala se han realizado eventos de olimpiadas científicas interuniversitarias, en las cuales, su mayor fuerte siempre ha sido la matemática y la física.

En el 2013, se tomó la decisión de implementar las primeras olimpiadas científicas interuniversitarias, coordinadas por la Escuela en Ciencias y Sistemas de la, Facultad de Ingeniería, Universidad de San Carlos de Guatemala.

Para realizar este evento y poder tener un control total sobre el mismo, se debe utilizar una herramienta de administración de eventos de programación. Se tomaron en cuenta varios sistemas implementados de código abierto, debido al poco tiempo que se posee para la implementación de este sistema, se decidió utilizar una herramienta de código abierto que cubriera parcialmente las necesidades del evento e implementar los cambios sugeridos por los usuarios.

Se realizaron las modificaciones al sistema y se creó un sitio en línea para que pueda ser accedido desde cualquier lugar con conexión a internet.

OBJETIVOS

General

Organizar y administrar el primer evento de Olimpiadas Tecnológicas Interuniversitarias realizadas por la Universidad de San Carlos de Guatemala a nivel nacional.

Específicos

1. Aportar ideas para el protocolo del evento.
2. Aportar en la creación de los problemas de programación utilizados en el evento.
3. Crear una herramienta para la administración del evento de programación.
4. Sentar las bases para futuras olimpiadas de tecnología.
5. Otorgar el código del sistema y manuales para su modificación en el futuro.

INTRODUCCIÓN

La Universidad de San Carlos, desde sus comienzos ha promovido el uso de la ciencia y la tecnología en la enseñanza que dirige a miles de jóvenes a lo largo del país, creando eventos extraaula tales como talleres y conferencias, con figuras distinguidas del país, sino también con la invitación de personajes internacionales, lo cual enriquece más la enseñanza.

Desde ya hace un tiempo, la Facultad de Ingeniería hace su aporte a promover la ciencia y la tecnología, por medio de olimpiadas científicas, en las cuales los jóvenes pueden mostrar sus conocimientos de la materia y mejorar al encontrar rivales de gran talla.

En 1986 fue celebrada la primer Olimpiada Nacional de Física, Química y Matemáticas, en las que participaron la Universidad del Valle (UVG) y la Universidad de San Carlos (USAC) iniciando así un legado que continua en la actualidad, añadiéndose también a las categorías, la de ciencias y sistemas.

1. FASE DE INVESTIGACIÓN

1.1. Antecedentes de la empresa

“La carrera de Ingeniería en Ciencias y Sistemas a nivel de licenciatura fue creada por la Junta Directiva de la Facultad de Ingeniería en su sesión del 25 de septiembre de 1970, según acta No. 819, punto cuarto. La misma empezó a funcionar en 1972, y en ese entonces se definió como: La Ingeniería en Ciencias y Sistemas es aquella rama de la ingeniería que trata con los problemas característicos en la planificación, diseño, operación y control de los sistemas en gran escala que tienen su origen con el desarrollo tecnológico de la sociedad”¹.

1.1.1. Reseña histórica

La naturaleza de los problemas con que se enfrenta el ingeniero en sistemas lo lleva a integrar equipos de trabajo en donde participan profesionales de otras áreas. El enfoque interdisciplinario que caracteriza a este profesional hace necesario que conozca con profundidad las ciencias de la ingeniería y que tenga suficiente facilidad en la aplicación de los métodos matemáticos, especialmente en lo que se refiere a los modelos del mundo físico.

De esta cuenta en 1977 se modificó temporalmente la carrera por un programa de posgrado con el nombre de Maestría en Sistema, dirigido principalmente a aquellos profesionales con títulos de Ingeniería, Arquitectura,

¹ Escuela de Ciencias y Sistemas.
https://ecys.ingenieria.usac.edu.gt/index.php?option=com_content&view=article&id=47&Itemid=57. Consulta: 12 de febrero 2014.

Economía o Administración de Empresas, que por la naturaleza de sus trabajo requerían de profundizar en las nuevas técnicas que ofrecía el enfoque de sistemas. Este programa pretendía que los estudiantes no ingenieros adquirieran los conocimientos de matemática y programación de computadoras equivalente a los que egresaban con el grado de licenciatura de la Facultad de Ingeniería.

En 1982 la Junta Directiva nombró una comisión de profesionales cuyo objetivo fue determinar el marco conceptual, necesidades y enfoque para la enseñanza de la computación y las técnicas del enfoque de sistemas en la Facultad. Como primera limitación, el propio marco e intereses de la Facultad de Ingeniería presentaba como única opción montar una carrera con un enfoque de ingeniería. Esto eliminó la posibilidad de implantar una carrera de sistemas de informática con un enfoque gerencial y/o administrativo, asimismo, quedó eliminado cualquier enfoque de Ciencias de la Computación que no lleve consigo los elementos de ingeniería.

Finalmente se optó por una carrera con el enfoque de Ingeniería en Computación, sin dejar por un lado los conocimientos del área de metodología de ciencias y sistemas, para los cuales esta carrera deberá actuar como vehículo de introducción y transmisión dentro de la Facultad de Ingeniería. Como consecuencia de lo anterior, la Junta Directiva de la Facultad de Ingeniería, en sesión del 9 de noviembre de 1982, en el acta No. 056-82, punto sexto, decidió reestructurar el *pensum* de estudios de la carrera, haciendo ver la necesidad de brindar a los estudiantes guatemaltecos la oportunidad de estar al día en los últimos adelantos científicos, que es necesario el desarrollo académico en el área de procesamiento de información y de los sistemas. En ese momento se denominó Ingeniería de Computación y Sistemas, nombre que

por razones legales no fue posible utilizar, quedando hasta la fecha el título de Ingeniería en Ciencias y Sistemas y dejando la duración de esta en 5 años².

1.1.2. Misión

“Al estudiante otorgar las competencias acertadas que garanticen el éxito en la búsqueda del conocimiento por medio de los distintos estilos de aprendizaje y fomentando la investigación de manera permanente que le permita una mejor continuidad en su calidad de vida. Tomando en cuenta las opciones que el país ofrece a las distintas áreas del mercado actual (logística, administración, información, tecnología, finanzas, contabilidad, comercial, etc.), tomando en cuenta el ámbito internacional debido a la alta competencia que se manejan en estos tiempos.”²

1.1.3. Visión

“Reconocer al estudiante de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala como un profesional de alto nivel, en base a los saberes incorporados en el *pensum* de estudios que permitan formar al estudiante de manera integral para el ejercicio profesional otorgándole los instrumentos adecuados para su desarrollo ocupacional.”³

² Escuela de ciencias y sistemas.
https://ecys.ingenieria.usac.edu.gt/index.php?option=com_content&view=article&id=47&Itemid=57. Consulta: 25 de febrero 2014.

³ Ibid.

1.1.4. Servicios que realiza

“El perfil del egresado de la carrera de Ingeniería en Ciencias y Sistemas de la Universidad de San Carlos de Guatemala, data de 1994. Está dividido por áreas de acuerdo a los aspectos a considerar dentro de las habilidades y conocimientos que deben llenar los egresados.

Conocimientos: de índole general, en el que se consideran las ciencias académicas generales, relaciones interpersonales e idioma inglés. De índole específica, referentes a las ciencias de su especialidad, que cubren las ciencias de la computación, metodología de sistemas y desarrollo de sistemas de información.

Habilidades: planteo y resolución de problemas, visión integral de los problemas considerando todos los elementos en juego, utilizando un enfoque sistemático”.⁴

1.2. Descripción de las necesidades

Las olimpiadas científicas organizadas por la Universidad de San Carlos de Guatemala y compuestas por personas o equipos de otras universidades han tenido auge desde su creación, por lo que se consideró agregar a esta competición el área de tecnología específicamente del área de programación de sistemas de cómputo.

⁴ Escuela de ciencias y sistemas.
https://ecys.ingenieria.usac.edu.gt/index.php?option=com_content&view=article&id=47&Itemid=57. Consulta: 25 de febrero de 2014

Por lo anterior y para una mejor administración de esta clase de competición se necesita implementar una herramienta capaz de administrar las presentes y futuras olimpiadas tecnológicas, lo suficientemente flexible que su configuración pueda realizarse de forma sencilla.

Tabla I. **Necesidades**

Nombre	Descripción
Problemas	Propuestas de problemas de programación para los participantes de las olimpiadas y la definición de la cantidad de estos, además de la ponderación de los mismos.
Protocolo del evento	Propuesta de los pasos a realizar en el curso del evento, como el inicio del evento, las palabras por los organizadores, los requisitos mínimos que debería de tener cada equipo.
Inscripción y conformación de equipos	Impresión de fichas de información y distribución en las universidades que participarán, se especificó que los equipos debían ser conformados por 4 integrantes, nombrando un coordinador por equipo e indicando el lenguaje y el IDE a utilizar.
Especificaciones técnicas	Definición de las necesidades de infraestructura para el correcto funcionamiento de la herramienta, hardware, cantidad de usuarios a utilizar la herramienta y sistema operativo en el cual se utilizará.
Especificaciones de la herramienta	Lenguaje de codificación, especificación de funcionalidad, configuración dinámica de la herramienta, plataforma de programación, alcances y límites de la herramienta.

Fuente: elaboración propia.

1.3. Priorización de las necesidades

Las necesidades serán priorizadas según un calendario de entrega como el siguiente:

Tabla II. **Priorización de necesidades**

Fecha	Actividad
17 de agosto de 2013	Protocolo del evento
24 de agosto de 2013	Inscripción y conformación de equipos
7 de septiembre de 2013	Problemas
21 de septiembre de 2013	Especificación de la herramienta
28 de septiembre de 2013	Especificaciones técnicas

Fuente: elaboración propia.

2. FASE TÉCNICO PROFESIONAL

2.1. Descripción del proyecto

El proyecto, según especificaciones de la Escuela de Ciencias de la Facultad de Ingeniería, consiste en contribuir en el desarrollo del primer evento de olimpiadas olímpicas interuniversitarias, coordinada por la Universidad de San Carlos de Guatemala.

El proyecto se divide en dos fases, las cuales se definen como: fase de preparación del evento y la fase de implementación de la herramienta de administración.

2.1.1. Preparación del evento

Para la preparación del evento se realizó un análisis de lo necesario para la ejecución del mismo, lo más básico e indispensable para que el evento se realizará.

2.1.1.1. Problemas de programación

Los problemas de programación se determinaron según estándares internacionales de eventos de igual índole, en universidades de Estados Unidos y por eventos que Google organiza año con año.

Todos los problemas planteados para el desarrollo del evento, fueron pensados para que no solo estudiantes de Ingeniería en Sistemas fuesen capaces de resolverlos, al contrario, todos fueron planteados para que la parte más importante del desarrollo del problema fuese el análisis del mismo, por lo que se permitió que estudiantes de cualquier facultad participaran.

Los problemas se presentan en 3 niveles, los cuales van desde los más sencillos, en los cuales no se necesita un nivel alto de análisis, problemas de nivel medio, en los cuales el nivel de análisis se incrementa, y por último los problemas de alto nivel, los cuales requerirán de un alto nivel de análisis y complejidad de programación.

2.1.1.2. Ponderación de problemas

Dependiendo de la complejidad de los problemas es la ponderación de estos.

Tabla III. **Ponderación de problemas**

Nivel	Ponderación
Bajo	5 unidades
Medio	De 10 a 20 unidades
Alto	De 25 a 45 unidades

Fuente: elaboración propia.

2.1.1.3. Inscripción de grupos

Al momento de la inscripción de cada grupo, se especificó que estos deberían ser de 4 integrantes, entre los cuales uno de ellos debería de actuar como coordinador. Cada uno de los grupos deberá especificar el lenguaje de programación que utilizarán, así como el IDE y el sistema operativo. Con la información anterior, se tendrá una computadora por grupo en el lugar del evento con las especificaciones otorgadas por los grupos.

2.1.1.4. Protocolo del evento

El flujo del evento se define en los siguientes pasos, según protocolos estudiados de eventos en Estados Unidos y olimpiadas científicas realizadas en la Universidad de San Carlos de Guatemala. El evento durará 2 horas, en las cuales se realizará lo siguiente:

- Ingreso a las instalaciones: los grupos deben presentarse por lo menos 15 minutos antes del inicio del evento, con sus integrantes completos.
- Organización de los grupos: los grupos serán sitiados en las computadoras preparadas para cada uno de ellos, con las especificaciones otorgadas por ellos y se les definirá un número de grupo para ser identificados.
- Presentación de autoridades: los organizadores y jueces del evento son presentados, así como las reglas del evento y son repartidos los problemas a cada grupo.
- Presentación de la herramienta: al momento en el cual los organizadores son presentados, los administradores del sistema, indican el correcto uso de la herramienta para cargar las respuestas de los problemas.

- Inicio del evento: ya que se anuncian las reglas y la forma de cargar las respuestas, se inicia el evento, los grupos pueden aclarar dudas de redacción de los problemas únicamente.
- Finalización del evento: después de transcurrir las 2 horas, los grupos deben dejar sus actividades, los jueces califican los problemas por medio de la herramienta y dependiendo de los aciertos y la ponderación de los problemas resueltos se concluye el grupo ganador.

2.1.2. Herramienta de administración

La herramienta de administración debe ser una solución liviana, confiable y portable con las siguientes opciones:

- Administración de eventos
 - Configuración de título, subtítulo del sitio y tiempo del evento.
 - Administración de usuarios y roles sobre el sistema.
 - Administradores
 - Jueces
 - Equipos
 - Configuración de Lenguajes de programación a utilizar
 - Configuración de problemas a evaluar
 - Descripción
 - Archivos de entrada
 - Archivos de salida
 - Ponderación
 - Configuración de penalizaciones
- Administración de Juzgado
 - Listado de problemas a ser juzgados

- Pantalla de auto evaluación de archivos de salida
- Administración de equipos
 - Pantalla de carga de archivos de salida por equipo
- Administración de puntuación del evento
 - Pantalla de visualización de puntaje y posiciones de los equipos en el evento.

2.2. Investigación preliminar para la solución del proyecto

Para implementar la herramienta de administración del evento, se realizó una investigación de herramientas y eventos similares para tener una idea más específica de lo que se espera al finalizar este evento.

2.2.1. Concurso de programación

Existen varios eventos de programación alrededor del mundo, de los cuales puede estudiarse el protocolo que utilizan, además de los problemas, las ponderaciones y el tiempo para cada uno de ellos. A continuación muestran algunos de las más destacadas y que han sido de mayor importancia en el desarrollo del proyecto:

Tabla IV. **Programming Contest**

Nombre	Descripción
International Conference on functional programming	Sus siglas en inglés ICFP. Celebrado desde hace una década en junio o julio de cada año. Con sede en Alemania, cualquiera puede ingresar, con cualquier lenguaje de programación en cualquier locación, es gratis y los equipos no tienen límite de integrantes.
The BME International	Es una competencia internacional llevada a cabo en Europa una vez por año para grupos de 3, en la cual los concursantes deben llevar sus computadoras.
International Collegiate Programming Contest	Uno de los más antiguos, iniciando en 1970 en Texas A&M, gestionado por la ACM desde 1989 y cuenta con la participación de IBM desde 1997. Uno de los concursos más grandes con miles de equipos procedentes de universidades y colegios que compiten a nivel local, regional y del mundo entero. Se deben realizar 8 problemas en un plazo de 5 horas.
The Obfuscated C contest	Este concurso se ha realizado por casi 20 años. Realizado por medio de entregables por correo electrónico. Se deben realizar programas en ANSI C en menos de 4.096 caracteres de longitud.
The ImagineCup	Estudiantes universitarios compiten codificando código aplicable a un conjunto de temas, por ejemplo el tema del 2008 fue "Imagine un mundo en donde la tecnología permita un ambiente sostenible"
Google Code Jam	En funcionamiento desde el 2008, abierto para personas de 13 años en adelante y personas que no trabajen para Google.

Fuente: <http://cplus.about.com/od/glossary/a/ten-contests.htm>. Consulta: 10 de enero de 2014.

2.2.2. Software de administración de eventos

Para la implementación de la herramienta, debido a limitaciones de tiempo, se tomaron en cuenta varias soluciones de código abierto para utilizar como base en el avance del proyecto. A continuación se muestra un cuadro comparativo de herramientas de administración de eventos de programación:

Tabla V. **Programming Contest Software**

Nombre	Descripción	Pro	Contra
DOMjudge	Es un sistema de juicio automático de eventos de programación. Se enfoca en la usabilidad y la seguridad.	Código abierto Documentación Cambios de versión recientes. Con licencia GNU	Es necesario tener un servidor Linux y además poseer permisos de administrador
CSUS Programming Contest	Es un sistema de control de eventos de programación, sus siglas en inglés PC ² . Desarrollado en la universidad del estado de California	Documentación reciente, notificaciones de cambio en el código	Aplicación de escritorio, puede descargarse gratis, pero no es de código abierto
Mooshack	Es un sistema de administración de eventos de programación basado en web. Entre sus opciones básicas se encuentran: evaluación automática de código cargado, respuesta a preguntas de clarificación, reevaluación de código, entre otros.	Código abierto, documentación y licencia GNU	Necesita correr en Linux y la instalación es tediosa,
Hackzor	Programa modular para la administración de eventos de programación, utiliza el framework Django Web como frontend	Código abierto, soportado por google, documentación	Necesita servidores Linux y conocimiento en programación con Python
CMS –	Es un sistema distribuido para ejecutar y organizar eventos de programación.	Código abierto, documentación, basada en python	Necesita servidor Linux y la instalación es tediosa.
PCCS	Programming Contest Control System un sistema de administración de eventos de programación basado en .net utilizado en competiciones internacionales como el ACM International Collegiate Programming Contest	Código abierto, documentación, fácil de modificar, configurable	No posee mucha documentación

Fuente: páginas electrónicas de: DOMjudge, CSUS, Mooshack, Hackzor, CMS, PCCS

2.3. Presentación de la solución al proyecto

Después de realizar el análisis de requerimientos para el evento e investigar la forma adecuada de realizar el evento, se debe implementar la herramienta y la forma en la que se realizará el evento.

2.3.1. Metodología de trabajo (Scrum)

Se creó un calendario de trabajo el cual consistía de un mes de trabajo:

- 5 *sprints* que consisten en
 - Factibilidad de desarrollo del sistema
 - Requisitos del sistema y fase 1 del sistema
 - Implementación y pruebas del sistema
 - Fase final del sistema
 - Implementación del sitio en servidor web en internet
- Retroalimentación de información cada semana
- 3 roles
 - Product Owner: Escuela de Sistemas de la Universidad de San Carlos.
 - Desarrollador: Jorge Luis Arroyo Pérez
 - Stakeholders: Organizadores de las olimpiadas científicas
 - Administradores: ingenieros encargados de laboratorios en Universidad Rafael Landívar.

Finalmente se optó, por cuestiones de tiempo, utilizar la herramienta PCCS codificada en Visual Studio con el Framework 3.5, el cual utiliza una conexión a base de datos local (MDF). Para que esta opción fuese viable, se

utilizaron las versiones Express de Visual Studio 2012 y se creó un sitio con dominio en la página smarterasp.net en la cual se publicó el sitio.








El sistema consta de las siguientes funcionalidades:

- Administración de eventos
 - Configuración de título, subtítulo del sitio y tiempo que durará el evento.
 - Administración de usuarios y roles sobre el sistema.
 - Administradores
 - Jueces
 - Equipos
 - Configuración de lenguajes de programación a utilizar por los equipos.
 - Configuración de problemas a evaluar.
 - Descripción
 - Archivos de entrada
 - Archivos de salida
 - Puntuación
 - Configuración de penalizaciones
- Administración de juzgado
 - Listado de problemas a ser juzgados
 - Pantalla de auto evaluación de archivos de salida
- Administración de equipos
 - Pantalla de carga de archivos de salida por equipo
- Administración de puntuación del evento
 - Pantalla de visualización de puntaje y posiciones de los equipos en el evento.

2.3.2. Plan de trabajo

El plan de trabajo consiste en el siguiente cronograma en el cual se detallan las actividades realizadas:

Figura 1. **Cronograma de actividades**

		Nombre	Duración	Inicio	Terminado
1		Evaluación de herramientas	7 days?	9/09/13 8:00	17/09/13 17:00
2		Investigación de herramientas de universidades	7 days?	9/09/13 8:00	17/09/13 17:00
3		Investigación de herramientas de terceros	3 days?	9/09/13 8:00	11/09/13 17:00
4		Verificación de funcionalidad requerida por el evento	2 days	18/09/13 8:00	19/09/13 17:00
5		Implementación de cambios necesarios	9 days?	20/09/13 8:00	2/10/13 17:00
6		Pruebas de estres sobre el sistema	1 day	3/10/13 8:00	3/10/13 17:00
7		Implementación del evento	2 days?	4/10/13 8:00	7/10/13 17:00
8		Documentación del sistema	5 days	8/10/13 8:00	14/10/13 17:00

Fuente: elaboración propia.

El día 5 de octubre se dio lugar a la primer Olimpiada Interuniversitaria organizada por la Universidad de San Carlos de Guatemala, realizada en la Universidad Rafael Landívar, iniciando a las 8:00 y finalizando a las 11:00.

La plataforma fue incapaz de ser utilizada por medio de internet debido a las especificaciones que los organizadores de la Universidad Rafael Landívar, por lo que fue necesario instalarla en una computadora portátil con las siguientes especificaciones:

Tabla VI. **Especificaciones de hardware**

Especificación	Descripción
Marca	Macbook
Sistema Operativo	Lion Os
Procesador	Dual Core 2.6 MHz
Memoria Ram	6 Gb
Disco Duro	128 Gb
Maquina Virtual	VMWare Fusion
SO virtualizado	Windows Server 2012
Memoria Ram virtualizado	4 Gb

Fuente: elaboración propia.

2.4. Costos del proyecto

Para calcular los costos del proyecto se debe tomar en cuenta que es un proyecto relativamente pequeño y se debe realizar en un tiempo sumamente corto, por lo que se tuvieron que dedicar mas horas/hombre de las planificadas para llevarlo a cabo.

2.4.1. Presupuesto

Ya que se trabajo con proyectos de código abierto y sin licenciamiento, el presupuesto se redujo únicamente al costo de *hosting* del sitio en una página en internet, en este caso smarterasp.net.

Tabla VII. **Costo del sitio**

Recursos	Cantidad	Costo unitario	Subtotal
Sitio smarterasp	1	\$40.00	\$40.00

Fuente: elaboración propia.

Figura 2. Especificaciones del sitio

General Hosting Features	
Webspace (GB)	1GB
Number of Site / IIS Entry [?]	1
Domain Names Hosted [?]	1
Subdomains	-
Bandwidth	10GB
Static IP	-
FTP	1
Website Traffic Statistics	✓
Access to Raw Log Files	-
Web-Based Control Panel	✓
Domain Name	\$10/yr
Domain Whois Privacy	\$5/yr
Windows 2012 / 2008 Hosting Features	
IIS8 / IIS7 Manager Remote Access Allowed	-
Remote Delegation Supported, Web Deploy	✓
Visual Studio 2012/2010 Support	✓
WebMatrix Support	✓
Full Trust and Medium Trust Supported	✓

Continuación de la figura 2.

Switch Between Integrated and Classic Mode	✓
URL Rewrite Module 2	✓
Site Memory Allocation	-
Isolated Application Pool	-
Run Multiple .NET versions	✓
Schedule Tasks / Cron Jobs	-
# of Concurrent Connections	200
MSSQL & MySQL Features	
MSSQL 2012, 2008, 2005	1
SQL Data Hosted on Separate Servers	✓
SQL DBO Rights, Store Procedures	✓
Full Text Search	✓
SQL Reporting Service 2012 [?]	-
SQL Backup and SQL Restore	✓
Attach MDF File Tool	✓
SQL Management Studio Access Allowed	✓
Contained Database Authentication	✓
ASP.NET ReportViewer Control	✓
SQL C.E 4.0	✓
MySQL5	1
Remote Connect	✓
MS Access	✓
Email / Mailing List Features	
Smartermail Enterprise Edition [?]	-
Email Boxes	-
SMTP, POP3, IMAP	-
Webmail Access	-

Continuación de la figura 2.

Auto Subscribe and UnSubscribe	-
Mailing List Form Generator	-
Mailing List Webservice API	-
Mailing List Template Manager	-
Scripting Features	
Classic ASP	✓
ASP.NET 2, 3.5	✓
ASP.NET 4.0	✓
ASP.NET 4.5.1 & 4.5	✓
Change .Net Versions	✓
Virtual Directory Setup	-
ScriptMap Setup	-
Custom Error Page	-
MVC 5 & 4, 3, 2, 1	✓
ASP.NET 4.5 Websockets	✓
WCF 4.0	✓
Node.js [?]	-
Crystal Report	✓
WCF RIA Services	✓
MS SilverLight	✓
PHP 5.2.x & 5.4.x	✓
Python 2.x & 3.x	-
CGI/Perl CGI/Exe	-
IonCube	✓
Zend Optimizer	✓
WAP 1.1/2.0	✓

Fuente: http://www.smarterasp.net/hosting_plans. Consulta: 25 de agosto de 2013.

2.4.2. Recursos

- Humanos
 - Coordinador del proyecto: asesor y coordinador de la Escuela de Sistemas USAC.
 - Desarrollador: Jorge Luis Arroyo Pérez
 - Técnico: Jossemar Cordero, práctica final, escuela de sistemas.
 - Jueces: Ingenieros encargados de la Universidad Rafael Landívar.
- Materiales
 - Sitio comprado en smarterasp.net
 - PC personal para desarrollo de la aplicación
- Horas de trabajo
 - 200 horas distribuidas en reuniones con los coordinadores y adaptación de la herramienta.

2.5. Beneficios del proyecto

- Bases de protocolo para eventos futuros de la misma índole.
 - Actividades dentro del evento
 - Reglas
 - Generación de afiches para el evento
 - Inscripción de equipos
- Herramienta base de administración del evento
 - Código fuente de la herramienta
 - Manual técnico en el que se especifica el funcionamiento interno de la aplicación.
 - Manual de usuario, en el cual se explica la forma de configurar la aplicación.

3. FASE ENSEÑANZA APRENDIZAJE

3.1. Capacitación propuesta

La capacitación fue dividida en dos fases:

3.1.1. Capacitación del protocolo del evento

La capacitación del protocolo del evento fue impartida a los ingeniero encargados del laboratorio en la Universidad Rafael Landívar, en la cual se les transmitió la información por medio de correo, documentos escritos y de forma oral de cómo deberían llevarse las actividades del evento.

3.1.2. Capacitación de la herramienta

La capacitación de la herramienta fue realizada al técnico delegado por la escuela de sistemas, el cual apoyo el día del evento, así como al coordinador de la escuela de sistemas.

3.2. Material elaborado

- Manual de usuario para la correcta configuración de un evento en el sitio.
- Manual técnico conteniendo lo siguiente:
 - Diagramas de contenido del sistema
 - Sitio y configuración necesaria
 - Código fuente del sistema.

CONCLUSIONES

1. Una herramienta de administración de eventos es tan efectiva como la logística que se tenga organizada para el evento en sí, por lo que si el plan de actividades del evento falla, la herramienta quedará sin utilidad.
2. La implementación de un evento de esta magnitud a nivel nacional, hace que los jóvenes que participan se automotiven a ser cada día mejores profesionales por medio del reto que implica.
3. Al momento de incluir la tecnología y haciéndola parte de ciertos procesos, automatizándolos, reduciendo sus costes de tiempo, recursos materiales y de insumos humanos, puede percatarse que resulta de gran beneficio y otorga grandes aportaciones dentro del desarrollo de las actividades a mejorar.
4. Por consiguiente se tiene el objetivo y como fin primordial que al final de las etapas de desarrollo, pruebas e implementación de la herramienta web de administración, para llevar a cabo el primer evento de Olimpiadas Tecnológicas Interuniversitarias de la Universidad de San Carlos de Guatemala; se realizó de forma exitosa por medio de la evaluación de los equipos en el evento.

RECOMENDACIONES

1. Al utilizar una herramienta Web debe tenerse en cuenta que la computadora servidor en donde se implemente la solución tenga las siguientes especificaciones mínimas: procesador Dual Core o superior, por lo menos 8 Gb de RAM y un sistema operativo Windows Server Core ya que es gratuito.
2. Se debe dar mantenimiento a la herramienta por cualquier cambio que se desee realizar, ya que su funcionalidad es limitada en tanto a la evaluación de código, se debe tomar la decisión implementar dicha funcionalidad en la herramienta u optar por una herramienta más sofisticada.
3. Ya que la seguridad está basada en autenticación de usuarios por Windows, al momento de crear un nuevo usuario se debe verificar en la base de datos que todos posean el mismo Machine Number.

BIBLIOGRAFÍA

1. ACM ICPC. *International collegiate programming contest*. [en línea] <<http://icpc.baylor.edu/>> [Consulta: 30 de septiembre].
2. Code Chef. *Programming Language Training Plataform*. [en línea] <http://www.codechef.com/> [Consulta: 27 de septiembre]
3. Github. *Contest Maganger System*. [en línea] <<https://github.com/cms-dev/cms>> [Consulta: 4 de octubre].
4. PC2. *CSUS Programming Contest*. [en línea] <<http://www.ecs.csus.edu/pc2/pc2docs.html>> [Consulta: 2 de octubre].
5. SourceForge. *Programming Contest Control System*. [en línea] <<http://sourceforge.net/projects/pccs-umd/>> [Consulta: 28 de septiembre].

APÉNDICES

Propuesta de reglamento para olimpiada interuniversitaria en el área de tecnología

Objetivo

Promover entre los estudiantes de las universidades del país un reto interesante donde el ingenio y la innovación puedan expresarse a través del desarrollo de algoritmos para resolver problemas utilizando la programación de computadoras.

Reglas de esta competencia

1. **Equipos Participantes:** Los equipos deberán de estar constituidos por 4 participantes, no se podrá admitir equipos con más o menos participantes. Todos los integrantes de un equipo deberán ser estudiantes de la misma Universidad.
2. **Equipo Multidisciplinario:** Los estudiantes que conformen el equipo podrán pertenecer a la misma carrera, o bien, podrán ser de distintas carreras, siempre y cuando sean de la misma Universidad.
3. **Coordinador:** Cada equipo deberá nombrar un coordinador (está incluido en los 4 miembros del equipo), el coordinador será la persona responsable de entregar los problemas resueltos al jurado evaluador y es la persona a quien se indicará si la solución es o no correcta.
4. **Lenguaje de Programación:** El lenguaje de programación es abierto, la elección del mismo queda a discreción del equipo participante. El equipo deberá indicar el lenguaje de programación que desea tener instalado en el ordenador que se le entregara, así como el IDE que desea utilizar.
5. **Interfaz de Usuario:** Los algoritmos deberán de ser resueltos sin el uso de Interfaces Gráficas de Usuario. Los algoritmos serán ejecutados en línea de comandos.
6. **Sistema Operativo:** La elección del sistema operativo también queda a discreción del equipo participante. El equipo deberá de indicar que sistema operativo es el que desea tener en su ordenador para la competencia.

7. **Ordenador:** Los ordenadores entregados a los equipos tendrán las mismas características técnicas para evitar desventajas por el funcionamiento de un ordenador.
8. **Tiempo de prueba:** Se proporcionara un tiempo de 15 minutos a cada equipo para que inspeccione el equipo entregado para corroborar el correcto funcionamiento del mismo. Luego de este tiempo no se podrá realizar cambios al ordenador, instalaciones, etc.
9. **Dispositivos de Almacenamiento:** Ningún equipo podrá hacer uso de ninguna clase de equipo de almacenamiento, únicamente se permitirán los dispositivos provisto por los responsables de la competencia, entiéndase USB, CD, DVD, etc. En caso de necesitar instalar herramientas en el ordenador se permitirá hacerlo únicamente durante el tiempo de pruebas del ordenador y asesorados por el personal encargado de la competencia.
10. **Material de Apoyo:** Los equipos no pueden tener ningún material de apoyo, entiéndase documentos impresos, material digital, acceso a internet, etc.
11. **Faltas:** Los equipos que sean sorprendidos haciendo uso de material de apoyo, de cualquier tipo, serán penalizados con 5 puntos negativos. Si vuelven a ser sorprendidos serán descalificados de la competencia.
12. **Jurado evaluador:** El jurado evaluador estará formado por un representante nombrado por cada universidad que participe en la olimpiada y tendrá el apoyo de un estudiante de práctica final de USAC que brindará un soporte técnico a los equipos que participen en la competencia.
13. **Cantidad de equipos:** La cantidad de equipos que podrán inscribirse estará limitada por la capacidad del lugar donde se realizará la competencia y por los equipos de cómputo que estén disponibles para la misma.

Desarrollo de la competencia

Cada equipo debe estar inscrito debidamente en la competencia.

El día de la competencia deberán presentarse todos los integrantes del equipo, no se permitirá el cambio de integrantes o la falta de alguno de los integrantes inscritos. Se dará un tiempo justo para que los equipos estén

completos el día del evento, si pasado este tiempo el equipo no se encuentra completo quedara automáticamente descalificado.

Los equipos recibirán un documento impreso y en formato pdf en el cual se indicaran los problemas a resolver. Este documento contendrá 4 problemas escritos en inglés o español, cada uno de éstos estará valorado con puntos de dificultad.

Los equipos tendrán 2 horas para resolver el máximo de problemas posible.

Todos los problemas recibirán un archivo de texto como entrada, con los datos necesarios para su solución. Estos archivos tendrán un nombre determinado y se encontraran en una localidad especifica dependiendo del Sistema Operativo electo por el equipo.

El problema resuelto por los equipos deberá de generar un archivo de texto como salida en una ubicación determinada y con el nombre indicado en el documento de problemas.

Cuando un equipo termine un problema, el coordinador deberá indicarlo al jurado evaluador indicando el problema que han resuelto, seguidamente deberán entregar al jurado evaluador el archivo generado por el programa desarrollado, para su revisión. Los equipos deberán de continuar con la solución del resto de problemas mientras que los jueces revisan que el archivo entregado es una solución correcta al problema en cuestión.

Cada problema resuelto correctamente adjudicará al equipo con una cantidad de puntos que estará definida en el documento que contiene los

problemas, en caso de que 2 equipos tengan la misma cantidad de puntos, se tomará en cuenta el equipo que haya entregado más rápidamente las soluciones.

Adicionalmente, cada problema resuelto de forma incorrecta generará una penalización de 5 puntos negativos para el equipo.

Premiación

Se premiará a cada miembro del equipo que ocupe el primer lugar y se darán medallas a los miembros de los equipos que queden en los primeros 3 lugares.

Olimpiada interuniversitaria

Área de Tecnología

Octubre de 2013

Instrucciones: A continuación se le presenta una serie de problemas, cada problema tiene una valoración en puntos, debe tratar de realizar programas de computadora que resuelvan cada problema, puede resolver los problemas en cualquier orden deseado, al finalizar de resolver cada problema debe solicitar que sea validado con el archivo de prueba que le será entregado por los jueces, deberá generar su salida y entregarla para su verificación, si la verificación es correcta habrá obtenido los puntos en que se ha valorado el problema. Recuerde que el tiempo utilizado para resolver los problemas también es parte de la competencia.

Problema No. 1 (15 pts.)

Suppose you are reading byte streams from any device, representing IP addresses. Your task is to convert a 32 characters long sequence of '1s' and '0s' (bits) to a dotted decimal format. A dotted decimal format for an IP address is form by grouping 8 bits at a time and converting the binary representation to decimal representation. Any 8 bits is a valid part of an IP address. To convert binary numbers to decimal numbers remember that both are positional numerical systems, where the first 8 positions of the binary systems are:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Input

The input file, "IPAddress.in", will have a number N ($1 \leq N \leq 100$) in its first line representing the number of streams to convert. N lines will follow.

Output

The output file "IPAddress.out" must have N lines with a dotted decimal IP address. A dotted decimal IP address is formed by grouping 8 bit at the time and converting the binary representation to decimal representation.

Input Example:

```
4
00000000000000000000000000000000
00000011100000001111111111111111
11001011100001001110010110000000
01010000000100000000000000000001
```

Output Example:

```
0.0.0.0
3.128.255.255
203.132.229.128
80.16.0.1
```

Problema No. 2 (25 pts.)

SudokuX es una variación del juego Sudoku estándar. Similar a Sudoku estándar, en SudokuX se deben ingresar los números del 1 al 9 una vez en cada fila, columna y cuadrado de 3x3 dentro de la cuadrícula total de 9x9 del rompecabezas. Además, en SudokuX los números solo deben ocurrir una vez en cada una de las dos diagonales.

	1			4		6		
	2		5				3	1
4			3					
						3	9	
6				7				5
	7	3						
					4			2
8	6				2		5	
		7		8			1	

Cuadrícula total (9x9) de SudokuX

Entrada

La primera línea contiene T, el número de casos de prueba ($1 \leq T \leq 100$). Entonces, T casos de prueba siguen. Cada caso de prueba consiste en 9 líneas. Cada línea contiene 9 caracteres, los caracteres válidos son:

'.' : Representa una celda en blanco

'1'..'9' : Representa un valor del SudokuX en la celda que corresponde

Cada caso de prueba es separado por una línea en blanco. Además, le garantizan que cada rompecabezas de SudokuX dado tiene una solución única.

Salida

Para cada rompecabezas, la salida contiene la solución en el mismo formato que la entrada. Debe imprimir una línea en blanco después de cada caso de prueba.

Ejemplo de Entrada

2
.8....2..
.1....5..
..34..7..
..9.5....
.2...46..
3.....
9...2....
.....
.....4.7

....41...
...6....5
.....7.9.
....1.3..
.5.....1
.2.....
..18...76
.7.....2
.....3

Ejemplo de Salida

486715293
712938546
593462718
679251384
128394675
354876129
945627831
867143952
231589467

293541768
748692135
615387294
864715329
357269481
129438657
531824976
476953812
982176543

Problema No. 3 (15 pts.)

Los números que representan partes de un todo se denominan números racionales, fracciones o quebrados. En general, las fracciones se pueden expresar como el cociente de dos números enteros a y b:

$$\frac{a \quad (\text{numerador})}{\text{-----}} \\ b \quad (\text{denominador})$$

Una fracción está en su forma reducida o canónica si el numerador y el denominador no tienen un factor común. Por ejemplo, $\frac{6}{8}$ no está en su forma reducida pues ambos, 6 y 8, son divisibles por 2: $\frac{6}{8} = \frac{(2 \cdot 3)}{(2 \cdot 4)}$; sin embargo, $\frac{3}{4}$ es una fracción en su forma canónica.

Existen dos tipos de fracciones, propias e impropias. Una fracción propia es aquella en la que el numerador es menor que el denominador; $\frac{2}{3}$, $-\frac{7}{8}$ y $\frac{16}{19}$ son todas ellas fracciones propias. Una fracción impropia es aquella en que el numerador es mayor que el denominador; $\frac{3}{2}$, $-\frac{8}{4}$ y $\frac{7}{3}$ son fracciones impropias. Las fracciones impropias se pueden convertir en números mixtos o en enteros (por ejemplo, $\frac{3}{2} = 1 \frac{1}{2}$; $-\frac{8}{4} = -2$, y $\frac{7}{3} = 2 \frac{1}{3}$) si se divide el numerador por el denominador y el resto se expresa como una fracción del denominador.

Entrada

La primera línea contiene un número N de casos ($1 \leq N \leq 100$) y a continuación vienen N líneas con fracciones (pueden ser propias o impropias, y pueden estar o no en su forma canónica, NO vienen números mixtos), cada fracción se representa con un valor entero P (numerador) una diagonal '/' y un valor entero Q (denominador).

Salida

Por cada caso en la entrada, mostrará la fracción en su forma canónica y, en caso de fracciones impropias deberán ser expresadas como un número mixto, el formato será un valor entero S un espacio ' ' (solo en caso de que exista un componente entero) un valor entero P (numerador) una diagonal '/' y un valor entero Q (denominador).

Ejemplo de Entrada

7
6/8
3/4
2/3
-7/14
3/2
-8/4
7/3

Ejemplo de Salida

3/4
3/4
2/3
-1/2
1 1/2
-2
2 1/3

Problema No. 4 (25 pts.)

Blue and Orange are friendly robots. An evil computer mastermind has locked them up in separate hallways to test them, and then possibly give them cake.

Each hallway contains 100 buttons labeled with the positive integers {1, 2, ..., 100}. Button k is always k meters from the start of the hallway, and the robots both begin at button 1. Over the period of one second, a robot can walk one meter in either direction, or it can press the button at its position once, or it can stay at its position and not press the button. To complete the test, the robots need to push a certain sequence of buttons in a certain order. Both robots know the full sequence in advance. How fast can they complete it?

For example, let's consider the following button sequence:

O 2, B 1, B 2, O 4

Here, O 2 means button 2 in Orange's hallway, B 1 means button 1 in Blue's hallway, and so on. The robots can push this sequence of buttons in 6 seconds using the strategy shown below:

Time	Orange	Blue
1	Move to button 2	Stay at button 1
2	Push button 2	Stay at button 1
3	Move to button 3	Push button 1
4	Move to button 4	Move to button 2
5	Stay at button 4	Push button 2
6	Push button 4	Stay at button 2

Note that Blue has to wait until Orange has completely finished pushing O 2 before it can start pushing B 1.

Input

The first line of the input gives the number of test cases, T. T test cases follow ($1 \leq T \leq 100$).

Each test case consists of a single line beginning with a positive integer N ($1 \leq N \leq 100$), representing the number of buttons that need to be pressed. This is followed by N terms of the form "Ri Pi" where Ri is a robot color (always 'O' or 'B'), and Pi is a button position ($1 \leq Pi \leq 100$).

Output

For each test case, output one line containing "Case #x: y" (note that exists a blank space between ':' and 'y'), where x is the case number (starting from 1) and y is the minimum number of seconds required for the robots to push the given buttons, in order.

Input Example

```
3
4 O 2 B 1 B 2 O 4
3 O 5 O 8 B 100
2 B 2 B 1
```

Output Example

```
Case #1: 6
Case #2: 100
```