

**Instituto tecnológico de Costa Rica**

Escuela de Ingeniería en Computación

Campus tecnológico Central Cartago

Programación Orientada A Objetos

Grupo 02

Profesor: William Mata Rodríguez

Proyecto #1

Competencias de atletismo

Elaborado por:

Kendall Rodríguez Camacho c. 2022049438

Jocsan Adriel Pérez Coto c. 2022437948

Fecha de entrega: Jueves 20 de octubre del 2022

II Semestre 2022

## Contenido

Enunciado .....	3
Temas investigados .....	5
Software de control de versiones.....	5
Funciones de depuración.....	6
JavaDoc .....	6
Envío de correos.....	7
Revisión de cuenta de correos.....	7
Conclusiones.....	8
Lista de revisión y análisis de resultados .....	9

## Enunciado

### Competencias de atletismo

El siguiente proyecto es un programa con interfaz donde se simulan diferentes competencias de atletismo con sus diferentes pruebas y cada una con sus atletas de su respectiva competencia y prueba. De esta manera se cuenta con las siguiente funciones:

- **Atleta:**
  - Crear atleta
  - Modificar atleta
  - Eliminar Atleta
  - Lista de atletas
- **Disciplinas**
  - Crear Disciplina
  - Modificar Disciplina
  - Eliminar Disciplina
  - Lista de Disciplinas
- **Competencias**
  - Crear Competencia
  - Modificar Competencia
  - Eliminar Competencia
  - Lista de Competencias
  - Agregar una prueba a una competencia
  - Eliminar una prueba de una competencia
  - Agregar un atleta a una competencia
  - Eliminar un atleta a una competencia
- **Marcas de atletas**
  - Crear una marca
  - Modificar una marca

- Eliminar una marca
- Visores de atletas
  - Crear un visor
  - Modificar un visor
  - Eliminar un visor
  - Lista de visores
  - Seguir a un atleta
  - Enviar correos
- Lista de marcas
  - Competencia
  - Atleta
  - Mejores marcas del atleta
  - Prueba
- Configuración
  - Parámetro 1
  - Parámetro 2

## Temas investigados

### Software de control de versiones

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo de tal manera que sea posible recuperar versiones específicas más adelante. Los sistemas de control de versiones han ido evolucionando a lo largo del tiempo y podemos clasificarlos en tres tipos: Sistemas de Control de Versiones Locales, Centralizados y Distribuidos.

- **Sistemas de Control de Versiones Locales:** Los sistemas de control de versiones locales en vez de mantener las versiones como archivos independientes, los almacenaban en una base de datos.
- **Sistemas de Control de Versiones Centralizados:** Para facilitar la colaboración de múltiples desarrolladores en un solo proyecto los sistemas de control de versiones evolucionaron: en vez de almacenar los cambios y versiones en el disco duro de los desarrolladores, estos se almacenaban en un servidor.
- **Sistemas de Control de Versiones Distribuidos:** La siguiente generación de sistemas de control de versiones se alejó de la idea de un solo repositorio centralizado y optó por darle a cada desarrollador una copia local de todo el proyecto, de esta manera se construyó una red distribuida de repositorios, en la que cada desarrollador podía trabajar de manera aislada, pero teniendo un mecanismo de resolución de conflictos mucho más elegante que en su versión anterior.

## **Funciones de depuración**

En este caso se utilizaron dos funciones de depuración que están incorporadas con el IDE de NetBeans las cuales son el debugger y las funciones de creación de interfaces.

Para lo que seria el debugger, este ya es conocido por varias personas, la función de este es visualizar el proyecto mas a detalle o paso por paso, de esta forma podemos ver lo que va sucediendo.

Las funciones de creación de interfaces son ya implementadas por el IDE por lo que solamente para crear una interfaz se debe de crear en el mismo proyecto un JFrame, este seria una pantalla, ya luego podemos implementarle las características que se deseen.

## **JavaDoc**

Esta herramienta utiliza el formato de 'comentarios de documentos' para documentar las clases de Java. Los IDE como NetBeans tienen una herramienta JavaDoc incorporada para generar documentación HTML.

Un punto para tener en cuenta es que esta herramienta no afecta el rendimiento de la aplicación de ninguna manera ya que el compilador elimina todos los comentarios durante la compilación del programa Java.

Escribir comentarios en el programa y luego usar JavaDoc para generar la documentación es ayudar al programador / usuario a comprender el código.

La documentación HTML generada por JavaDoc es documentación de API. Analiza las declaraciones y genera un conjunto de archivos fuente. El archivo fuente describe campos, métodos, constructores y clases.

Tenga en cuenta que antes de utilizar la herramienta JavaDoc en nuestro código fuente, debemos incluir comentarios especiales de JavaDoc en el programa.

**Envío de correos**

Para el caso del envío de correos se tuvo que utilizar la librería de JavaMail, este es una librería que se debe de descargar desde internet para luego instalarla dentro del proyecto. Cuando este se descarga y se añade a nuestro proyecto nos permite usar todas las funciones necesarias para poder enviar correos desde java hasta un correo de Gmail.

**Revisión de cuenta de correos**

Para hacer la revisión de cuenta de correos nada mas tenemos que ingresar a los correos donde fue enviada la información.

## Conclusiones

El trabajo realizado fue un éxito, se pudieron resolver la mayoría de las indicaciones que el proyecto indicaba. Algunos inconvenientes que se encontraron en la elaboración del programa, pero se logró resolverlos investigando sobre el tema.

La elaboración de este trabajo fue de gran importancia para apañar y reforzar nuestros conocimientos sobre las interfaces gráficas en el IDE de NetBeans, así como el uso de clases en java, Además, gracias a este programa mejoramos considerablemente nuestros pensamientos lógicos al realizar algunos algoritmos que en un pasado no lográbamos resolver y que ahora los ejecutamos con gran facilidad.



## Lista de revisión y análisis de resultados

Concepto	Puntos Originales	Avance 100%/0	Puntos Obtenidos	Análisis de resultados
Configuración del sistema	2	100%	2	100: Totalmente desarrollado
Disciplinas y sus pruebas CRUD y listado	8	100%	8	100: Totalmente desarrollado
Aletas CRUD y listado	10	100%	10	100: Totalmente desarrollado
Competencias y sus pruebas CRUD y listado	8	100%	8	100: Totalmente desarrollado
Registro de marcas	8	100%	8	100: Totalmente desarrollado
Listado de marcas por competencia y prueba	8	100%	8	100: Totalmente desarrollado
Listado de marcas por atleta	8	100%	8	100: Totalmente desarrollado
Listado de mejores marcas por atleta en cada prueba	8	100%	8	100: Totalmente desarrollado
Enviar correos a visores	14	100%	14	100: Totalmente desarrollado
Listado de marcas por prueba	8	100%	8	100: Totalmente desarrollado
Validación de datos	8	100%	8	100: Totalmente desarrollado
Documentación del proyecto	10	100%	10	100: Totalmente desarrollado
<b>TOTAL</b>	100	100%	100	
Partes desarrolladas adicionalmente				

# Diagrama de Clases

