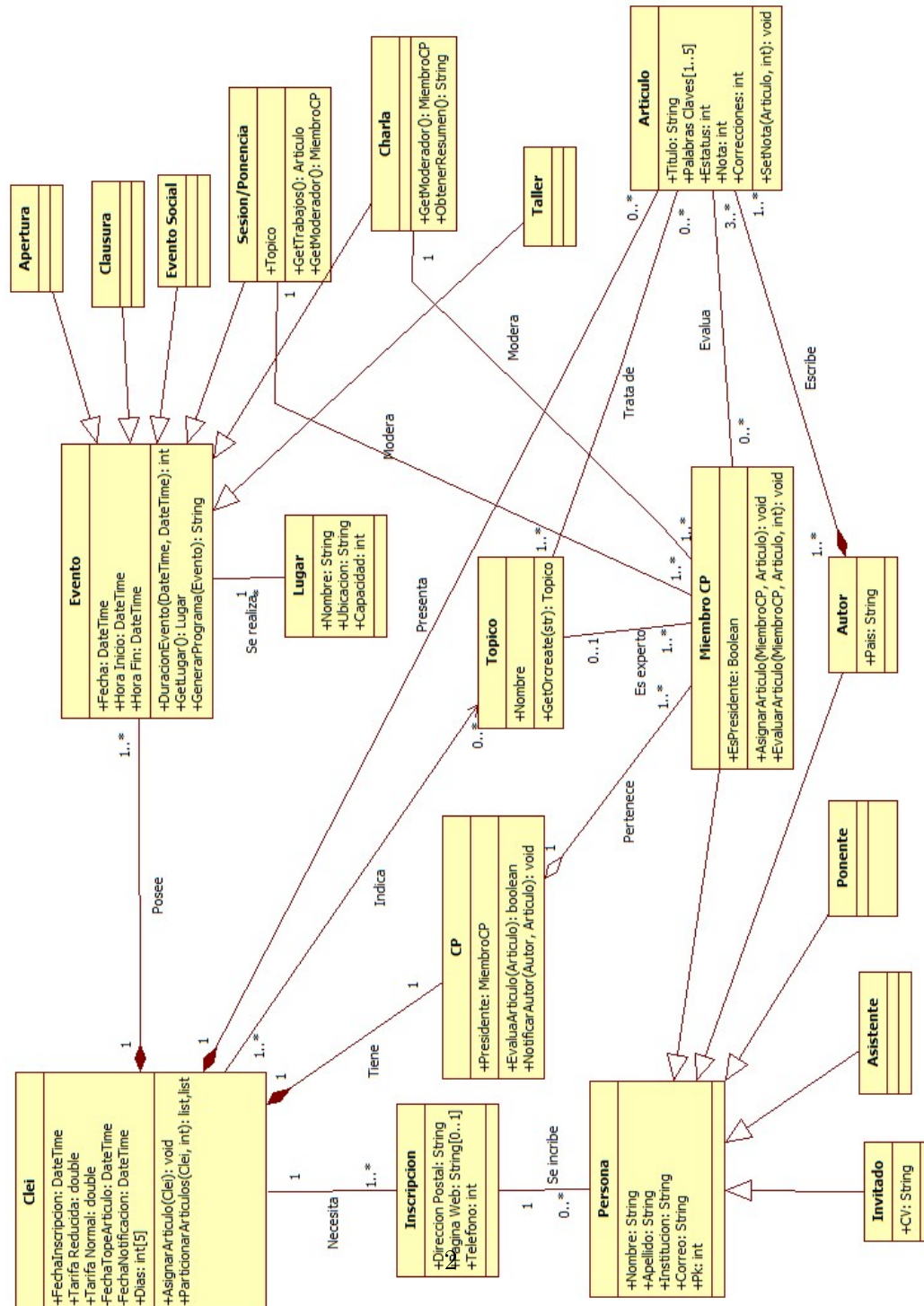


Informe Técnico

Daniela Rodríguez 09-10735 Daniel Rodríguez 07-41475

1 de diciembre de 2013

1. Diagrama actualizado



2. Reporte

Se debe tener instalando matplotlib y numpy para correr el programa puesto que se realizaron los histogramas solicitados.

El programa principal se debe correr *python main.py* . Adjunto se encuentra un archivo input.in el cual se puede copiar y pegar para probar el programa de forma rápida.

Para correr las pruebas, *python tests.py*.

2.1. Cambios realizados al diagrama

Se realizaron muy pocos cambios. El más notable, es agregar el atributo *id* a Persona, como forma de distinguir a diferentes personas que posean el mismo nombre.

También la inclusión del método *ParticionarArticulos* ya que hasta el momento de la entrega anterior no se manejaba el cómo se aceptaban los artículos.

Por último, a artículo, se le agregaron dos campos *nota* y *correcciones*, siendo este una ayuda para llevar la cuenta de cuántas correcciones posee.

2.2. Tiempo empleado

Para la realización de esta tarea se emplearon 15 horas, a lo largo de 5 días. Durante este período de tiempo, se rotó 6 veces.

2.3. Eclipse

Para esta entrega, decidimos utilizar el editor Emacs como IDE en lugar de Eclipse por cuestiones de eficiencia.

Aun cuando Eclipse es un IDE excelente para trabajar en Java, dado el corto tiempo para completar la tarea y el solapamiento con otras asignaciones (Sistemas de Información), resulta difícil trabajar efectivamente al tener que acostumbrarnos a un entorno de desarrollo distinto. Esto presenta un retraso al momento de comenzar a desarrollar el código.

Emacs, por otro lado, lo hemos venido usando desde ya hace tiempo y estamos familiarizados con él, por lo que pudimos enfocarnos por completo en la

asignación en lugar de estar aprendiendo como manejar un entorno diferente.

2.4. Casos de prueba y refactorización

Las pruebas se hicieron a medida que las clases fueron implementadas a manera de asegurar en lo posible su correctitud:

1. Persona
2. Tópico
3. Miembro CP
4. Autor
5. CP
6. Clei
7. Artículo

Cada vez que se realizó algún cambio considerable, las pruebas se volvieron a correr. Al momento de la entrega, ninguna prueba falló.

En cuanto a la refactorización, se realizó primero un prototipo general del programa, y después de lograr que el mismo funcionara, se procedió a refactorizar elementos que se repetían a lo largo del código en funciones generales. El caso más importante fue en las pruebas, donde se escribieron los mismos sets de prueba para cada clase, para luego darnos cuenta que se podían generalizar, de forma que solo era necesario implementar esta generalización para todas las pruebas y enfocarnos en los casos de uso particulares de cada clase implementada.