



UNIVERSIDAD SIMÓN BOLÍVAR
Departamento de Computación y Tecnología de la Información
CI-3715 Ingeniería del Software 1
Profs. Kristoffer Pantic & Alfonso Reinoza

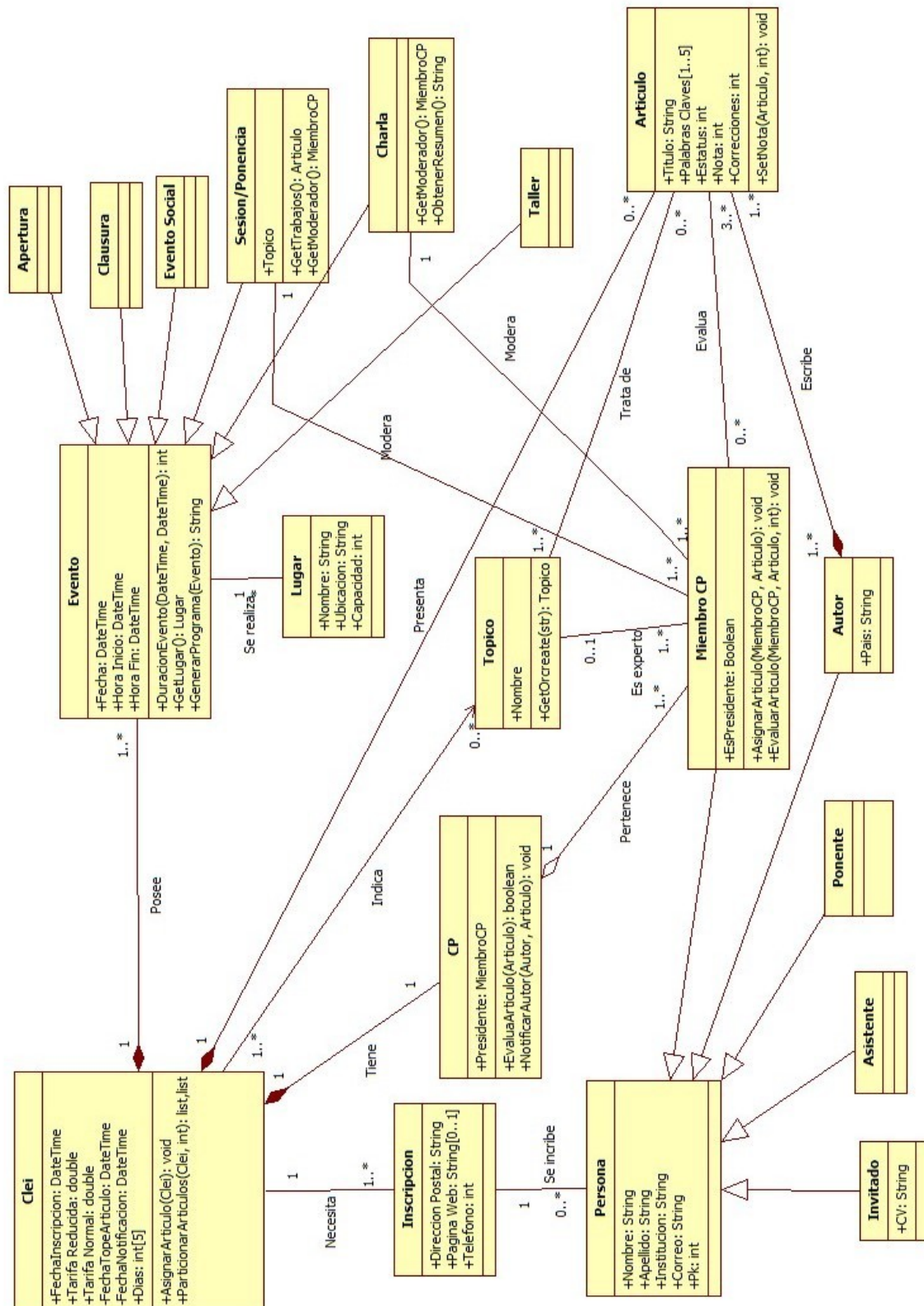
TAREA #4
Entrega final CLEI2014

Elaborado por:

José Delgado 09-10219
Daniela Rodríguez 09-10735
Samantha Campisi 10-10105
Carlos Cruz 10-10168
Esteban Oliveros 10-11252

Puntos a ser cubiertos en el presente informe:

1) Versión final del Diagrama de Clases



En este diagrama de clases se aprecian los conceptos más importantes para realizar el diseño de la aplicación de CLEI. En la implementación, llevada a cabo con la herramienta Django, figuran como aplicaciones (los diversos módulos) principales Personas, Evento, Artículo y Modulo_Clei. Estas fueron las clases principales, de las cuales heredan sus diversas subclases.

2) Reporte de Gestión de Riesgos

Hoy en día, ningún software que se desarrolle se encuentra exento de la presencia de riesgos que amenacen el éxito de su entrega. Así pues, al momento de desarrollar la aplicación de CLEI2014, el equipo se vio en la necesidad de identificar los posibles riesgos, analizarlos, elaborar planes para manejarlos y monitorizarlos. Se escogen 3 riesgos a corto plazo (ya que entra en el período de un sprint), que se listan a continuación:

a) Tecnológico

Incompatibilidad al momento de integrar el trabajo de cada miembro del equipo: debido a diferencias de sistema operativo y versiones de herramientas mientras se implementaban las funcionalidades. La probabilidad es alta (entre 50 y 75%) y posee un impacto catastrófico.

Este riesgo se mitigó a través de un establecimiento de estándares que seguir por parte de todo el grupo. A pesar de que no requiere ser monitorizado, no se descartó, ya que era de suma importancia unificar la tecnología a emplear para el desarrollo de la aplicación.

b) De equipo

Desmotivación del equipo: es una posibilidad real que se dio, ya que esta tarea representa la última entrega del trimestre y todos los miembros del grupo se encuentran cansados por todas las actividades llevadas a cabo a lo largo del período lectivo en varias asignaturas, lo cual compromete la rapidez del desarrollo. La probabilidad es muy alta (Más de 75%) y el impacto es serio.

El hecho de que sea la última entrega, quizás impulsa a dar un esfuerzo final. Requiere de monitorización para no desfallecer en el intento.

c) Asociado a la calidad del producto

Tiempo para realizar la entrega: es clásico que en un esquema de desarrollo ágil el tiempo sea una preocupación constante, ya que se trabaja para dar valor con cada entrega lo más rápido posible. El uso de Django como herramienta es nuevo para casi todo el equipo, por lo cual también requirió una inversión de tiempo considerable para aprender lo necesario.

La probabilidad es alta (50%-75%) con impacto catastrófico si no se maneja correctamente, por lo que se necesitó planificación bajo lo establecido en el modelo Scrum, con reuniones sobre las situaciones de cada miembro del equipo. Requiere de una constante monitorización.

3) Ventajas/Beneficios de los patrones probados

En el curso de Ingeniería del Software I se tocó el tema de patrones de diseño, ahondando en diversos tipos de patrones, cuyo fin principal es adecuar el producto que se entregará a la situación bajo la cual se debe desarrollar. Para esta última entrega, se hizo uso de dos patrones: Estrategia y Modelo-

Vista-Controlador, que se listan a continuación con sus beneficios.

a) Estrategia:

- Se factorizan aspectos comunes de una familia de algoritmos (por ejemplo, los de selección de artículos o escogencia de paquetes de inscripción) y se pueden utilizar en las clases base de la jerarquía.
- Proporciona mayor cohesión, pues cada estrategia tiene un único y claro propósito.
- Sistematiza el uso de implementaciones alternativas de la funcionalidad de aceptación de artículos y escogencia de paquetes de inscripción.
- Teniendo una única clase que haga uso de distintas estrategias, se reduce el uso de subclasses y, por tanto, el acoplamiento entre ellas.

b) Modelo-Vista-Controlador:

- Clara separación entre interfaz, lógica de negocio y de presentación.
- Sencillez para crear distintas representaciones de los mismos datos.
- Facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar desarrollo guiado por pruebas (TDD).
- Facilidad para desarrollar prototipos rápidos.

Cabe destacar que, a pesar de que en Django este patrón no se llame Modelo-Vista-Controlador, el Model-Template-View es análogo.

4) ¿Qué recomendar a la gerencia respecto al desarrollo del sistema del CLEI? / Próximos pasos a seguir

Es importante tomar en cuenta que esta aplicación permite crear un CLEI para cualquier edición, por lo que podría ser reutilizada en el año 2015. Los próximos pasos a seguir serán aquellos que den lugar a las pruebas de aceptación por parte de los clientes (usuarios) y la puesta en práctica del sistema.

En primera instancia, el sistema es muy sencillo de utilizar y funciona con perfiles (login). De este modo, únicamente se le permite a un administrador la creación de un nuevo CLEI, la elección de los miembros del Comité de Programa, agregar los lugares de la conferencia y los eventos que darán vida a la misma, así como sus moderadores.

Por otro lado, un autor o un participante pueden gestionar su proceso de inscripción, con las consideraciones establecidas en las previas entregas. Además, un autor puede agregar un nuevo artículo dentro del rango de fechas permitido, para que un miembro de CP pueda otorgarle una puntuación, y posteriormente un administrador elegirá la manera en la que se aceptarán los artículos ya puntuados.

Entre otras acciones elegibles, se pueden listar los participantes y todas las actividades que forman parte de la conferencia.