



Paradigmas de Programación 2022

Proyecto Final

Sistema de Validación de Comentarios

Integrantes:

Hidalgo Bini, Rocío

Heredia, Tomás

Nacchio, Javier Augusto

Ing. en Informática

Ing. en Informática

Programador Universitario

Contenido

1. Introducción	3
2. Requerimientos	4
3. Consideraciones del diseño	5
4. Desarrollo del trabajo	6

1. Introducción

Para culminar el cursado de la materia se planteó un proyecto grupal que abarque los temas aprendidos durante el mismo, entre los cuales destacamos la programación orientada a objetos, el diseño de clases, relación entre estas, agregación, composición, herencia, polimorfismo, entre otros.

En nuestro caso el proyecto consistió en un sistema de verificación de comentarios ofrecido por una compañía, para distintas empresas que contraten dicho servicio. Para esto, se requiere la información de estas empresas y de los usuarios habilitados a comentar sobre las mismas.



2. Requerimientos

Dicho sistema web requiere distintas funcionalidades que se detallan a continuación:

- Almacenar información de cada empresa que contrató el servicio (nombre, razón social, rubro, sitio web, redes sociales y administrador).
- Almacenar información de cada persona apta para comentar (número de teléfono, ciudad, provincia y país).
- Permitir al administrador de cada empresa la posibilidad de agregar comentadores a la empresa que administra.
- Permitir a estos administradores revisar cada comentario que realizan dichos comentadores, validándolos (o no) y finalmente elegir si serán mostrados (o no) en el sistema.
- Almacenar cada comentario realizado por cada comentador, con su fecha de creación y puntuación.
- Mostrar al usuario que califica positivamente a una empresa, las redes sociales de la misma.
- Clasificar por categorías a cada comentador.
- Reclasificar a cada comentador cada determinado tiempo.
- Calcular la valoración de cada empresa, en un intervalo de fechas, según el puntaje otorgado por cada usuario, modificado por el “peso” del puntaje otorgado debido a su clasificación.

3. Consideraciones del diseño

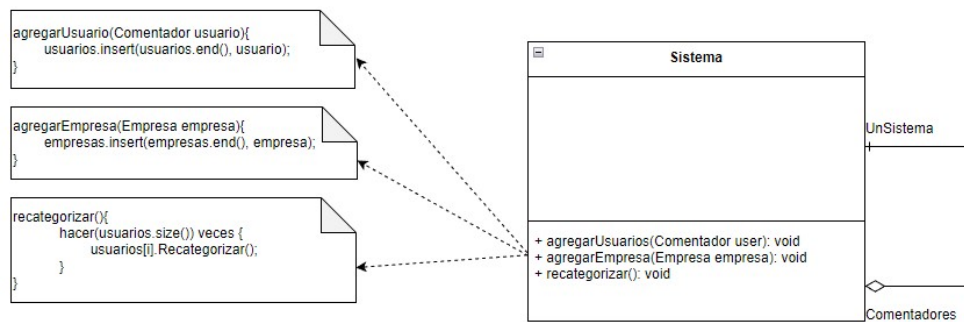
Para la elaboración del sistema en cuestión se tuvieron en cuenta distintas consideraciones que nombramos a continuación:

- Los comentarios se almacenan en el sistema. Esto es para facilitar la recategorización. -
- Cada comentador podrá comentar en distintas empresas. El enunciado lo pide así. -
- Cada administrador trabajará para una única empresa. Para simplificar el código. -
- Cada administrador podrá comentar en distintas empresas, con una cuenta de comentador. En primera instancia se pensó que los administradores y comentadores hereden de una clase persona, pero luego se cambió esto para que el administrador pueda comentar también. -
- Los comentarios serán guardados en la empresa a la que corresponden. Esto es para facilitar el acceso a los mismos y su puntuación. -
- Para la categorización de los comentadores se utilizarán clases. Esto permite que el método “calcularPeso” sea distinto para cada caso. -
- La validación consistirá en la lectura y visado de cada comentario por parte del administrador de cada empresa. Se llegó a esta conclusión junto a la profesora. -
- Primero se debe agregar una empresa sin administrador, luego crear el administrador para dicha empresa y finalmente asignar el administrador a esta. Esto es porque las clases Empresa y Administrador se conocen entre sí y esto implica hacer una declaración incompleta entre ambas para que el programa funcione. -

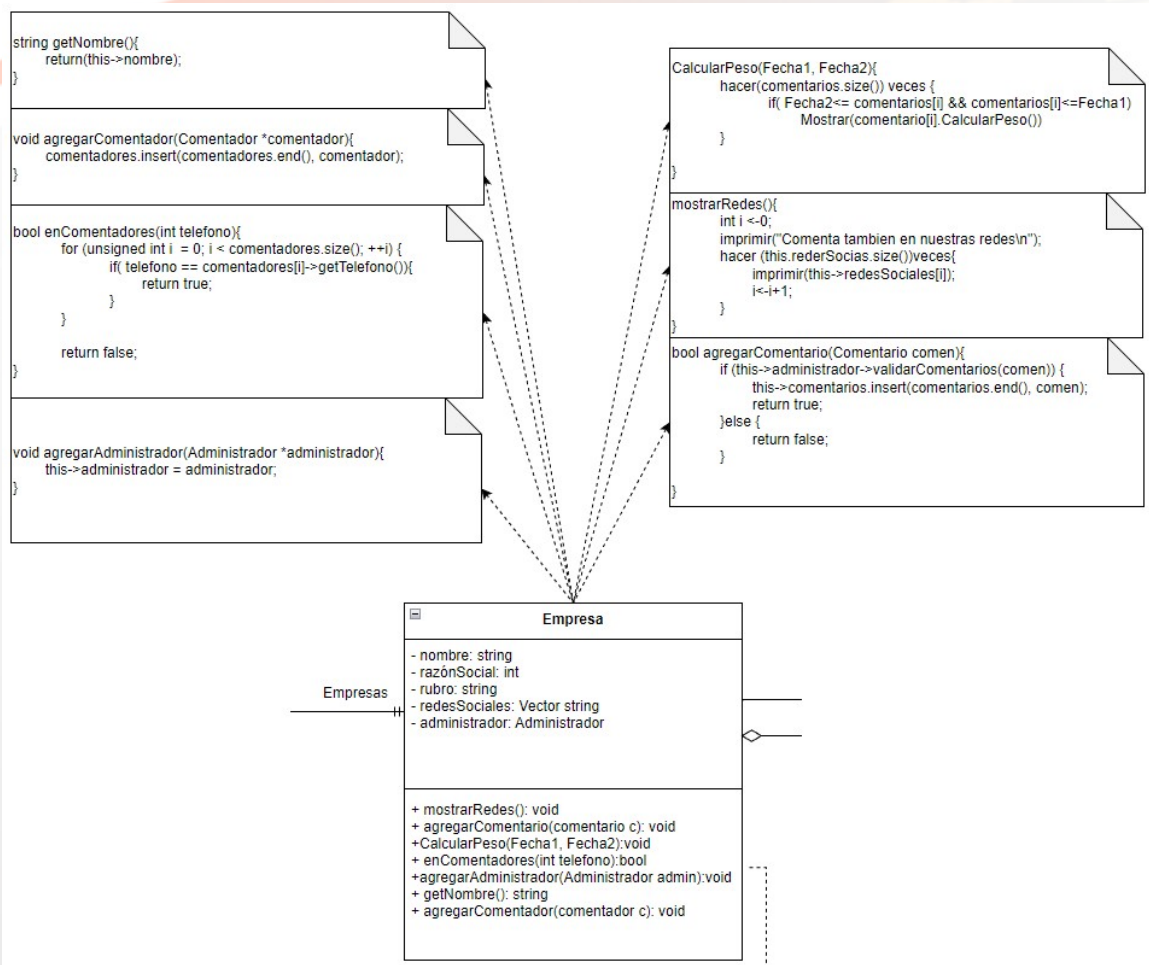
4. Desarrollo del trabajo

Etapa 1:

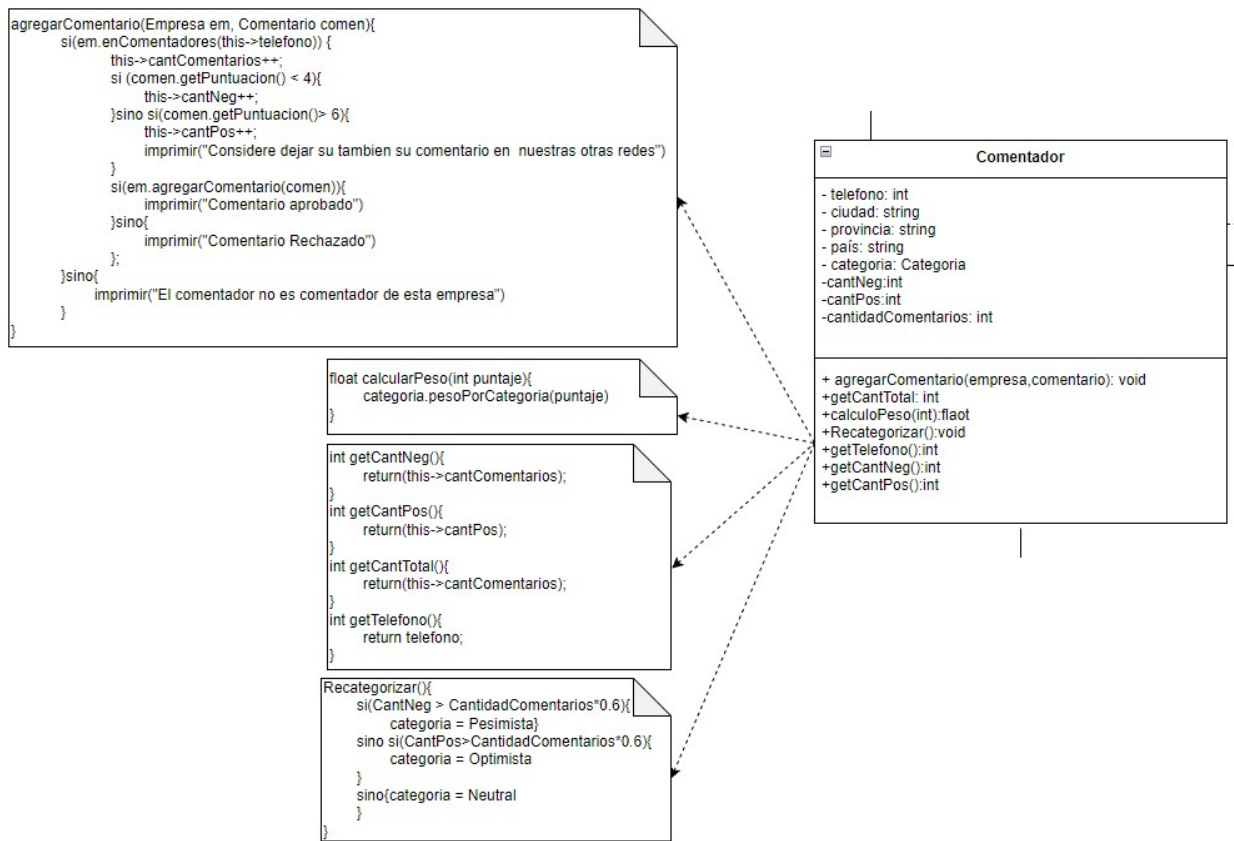
[Sistema]: Almacenar los comentadores, las empresas y es el encargado de recategorizar a los comentadores.



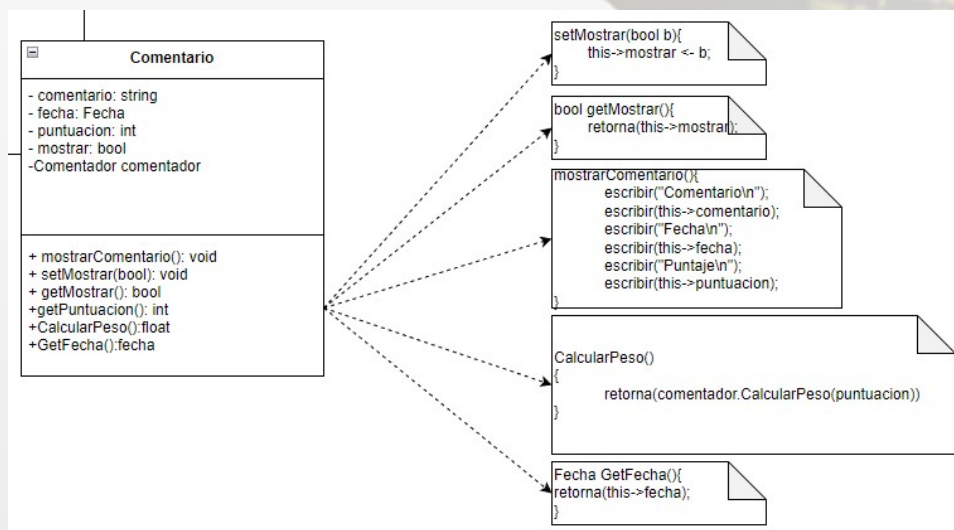
[Empresa]: Almacenar sus comentarios, calcular el peso de sus comentarios en un rango de fechas.



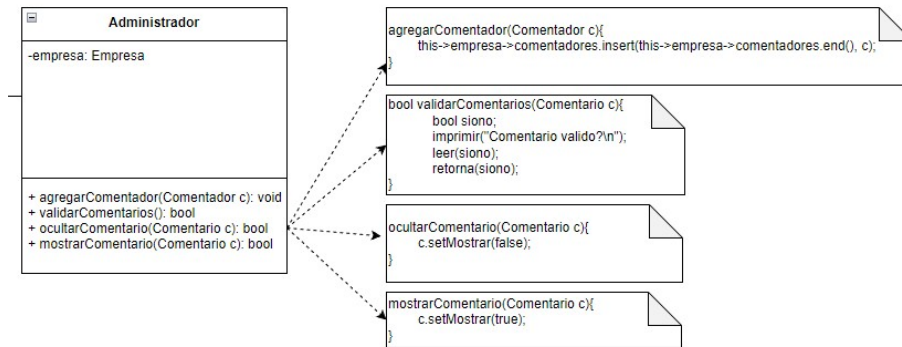
[Comentador]: Es el encargado de agregar comentarios. Posee una categoría que determina el peso de sus comentarios.



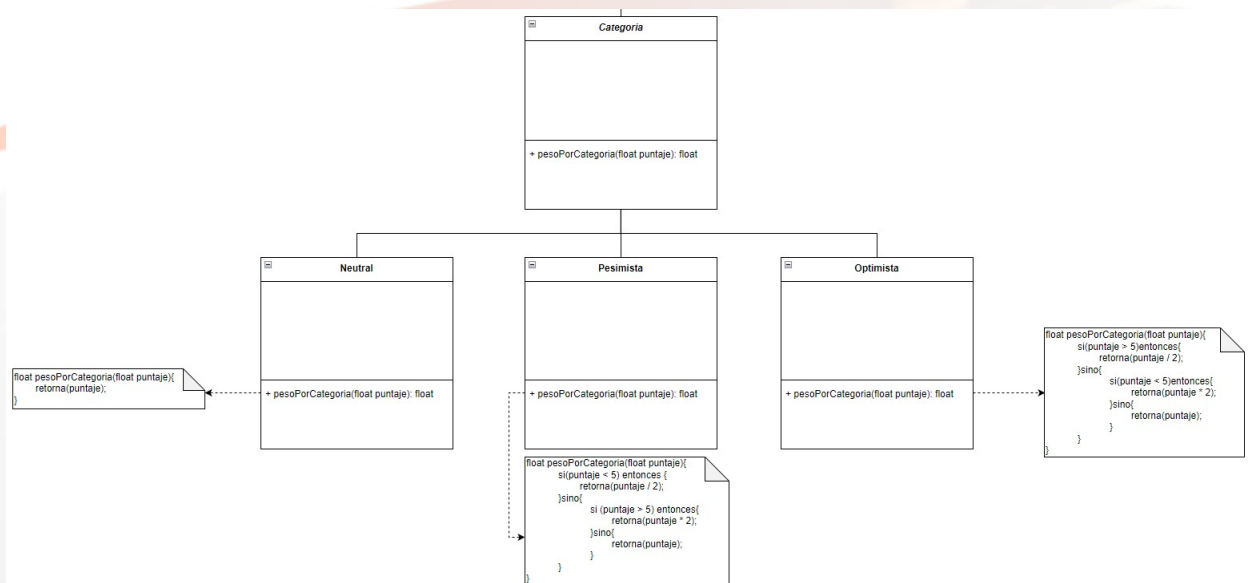
[Comentario]: Almacenar y mostrar los datos del comentario creado por el comentador.



[Administrador]: Es el encargado de revisar y aprobar los comentarios que realizan los comentadores. A su vez, agrega comentadores aptos para su empresa y modifica las propiedades de visibilidad de los comentarios.



[Categoría]: Es una clase virtual de la cual heredan las clases Neutral, Pesimista y Optimista y se encarga de calcular el peso de cada comentario.



Etapas 2:

Diagrama de Clases

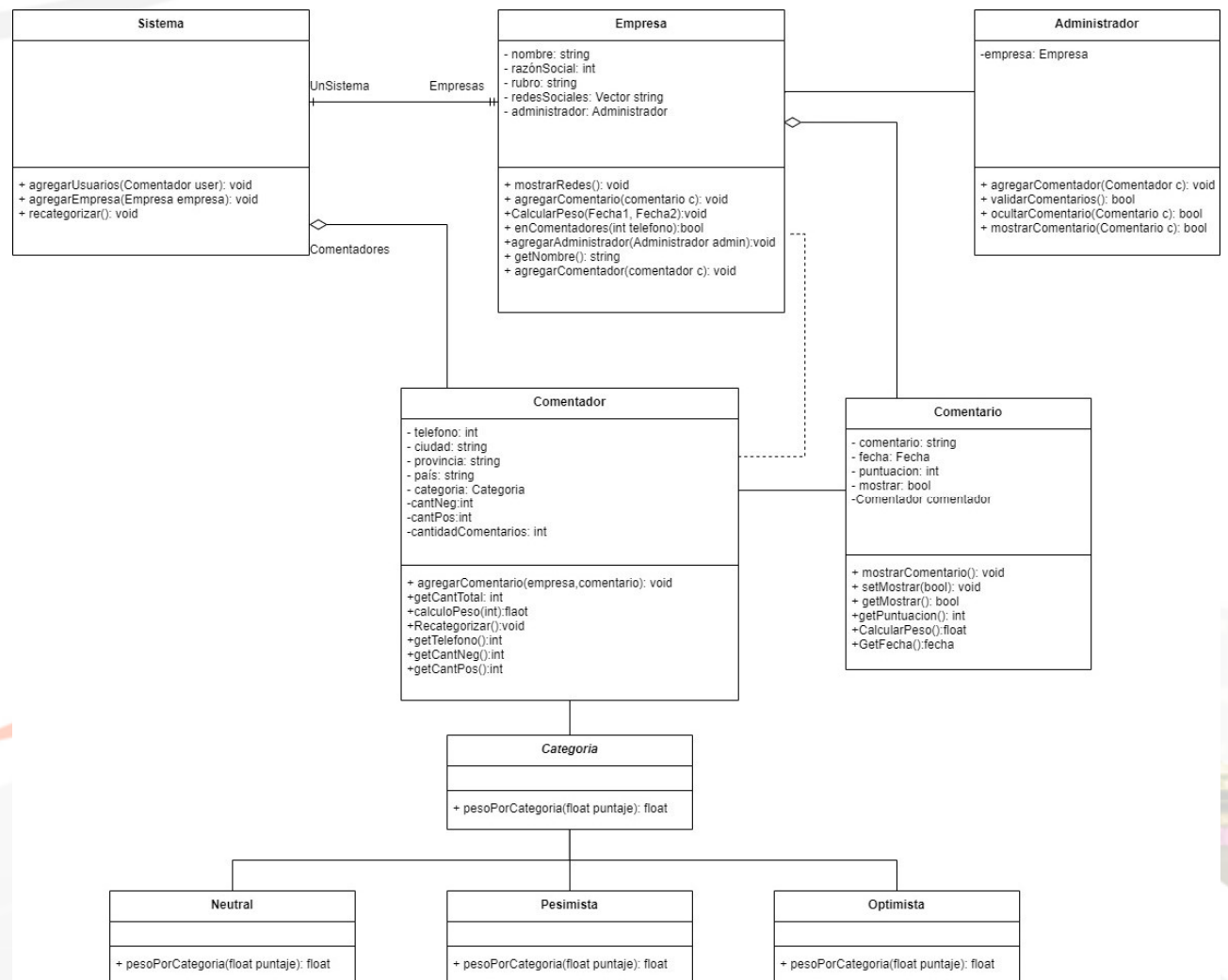
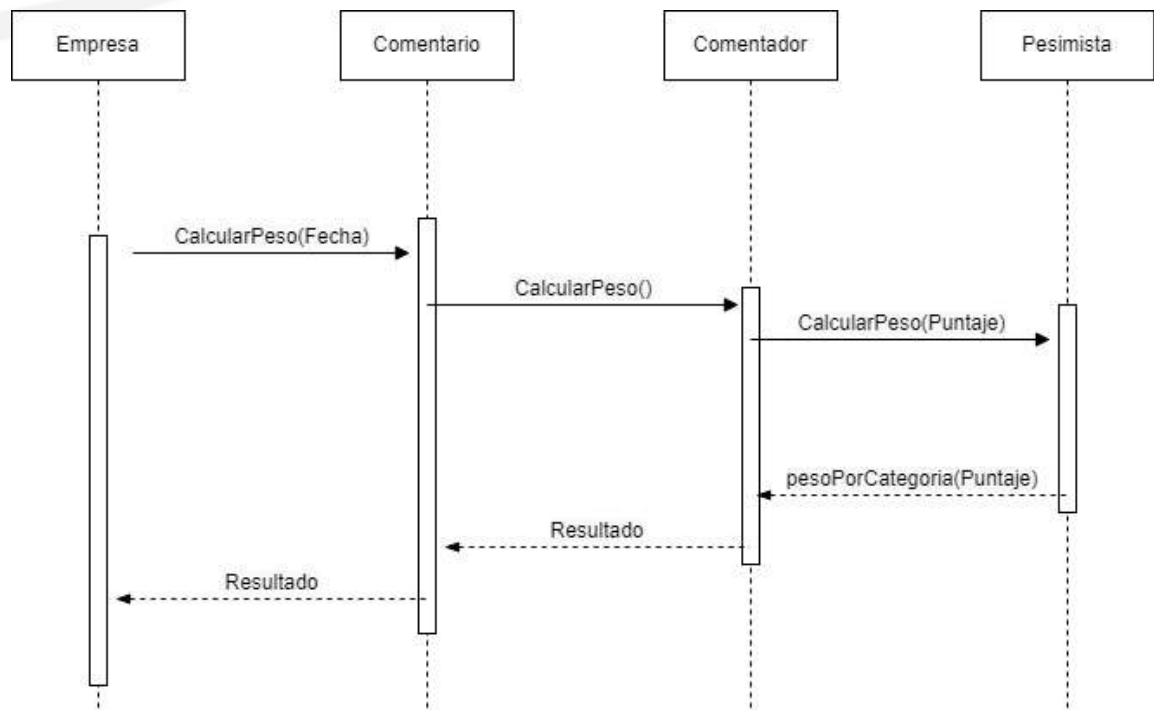


Diagrama de Secuencia de método de recategorización.



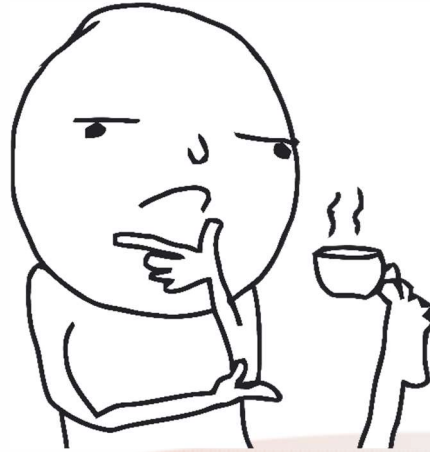
Etapas 3:

Consideraciones de Implementación:

- Se resolvió la relación entre Administrador y Empresa con una declaración incompleta en Administrador. -
- Primero se crea una empresa sin Administrador, luego el Administrador para esa Empresa y finalmente se lo agrega a la misma. -
- La clase Comentarador también se resolvió con declaración incompleta de Empresa. -
- El Comentarador recibe una Empresa y un comentario ya creado y se lo pasa a Empresa para que lo agregue a su lista de Comentarios. -
- El Administrador debe validar los comentarios por teclado. -
- El método recategorización se hace desde el Sistema cuando este lo desee y utiliza el mismo método en la clase Comentarador. -
- El método calcularPeso se llama desde Empresa y esta accede al mismo método de cada Comentario. -
- Se trabajaron las categorías como herencia para más sencillez. -
- En la clase Empresa, el vector de comentaradores es un vector de punteros a los comentaradores que ya están en el Sistema para optimizar espacio y velocidad. -
- Cuando se quiere agregar un comentario primero se verifica que su comentarador sea un comentarador de la empresa a la cual se quiere agregar dicho comentario.-
- Para la categorización de un comentarador consideramos que un puntaje es negativo si este es menor a 4 y es positivo si es mayor a 6, en otro caso en neutral. -
- En la clase Comentario se hace una declaración incompleta de Comentarador para que este tenga registro de quien lo creo. -

Lunes 7/11:

- Lectura del enunciado.
- Consulta con los profesores.
- Diferenciación entre clases y los métodos.



Miércoles 9/11:

- Desarrollo de la primera versión del diagrama UML.
- Desarrollo de la primera versión del código.



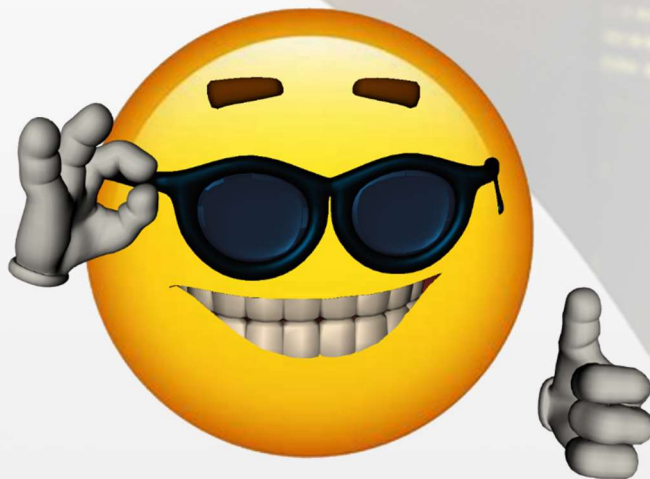
Lunes 14/11:

- Revisiones en el diagrama UML. Se cambian varias cosas.
- Desarrollo de la segunda versión del código.



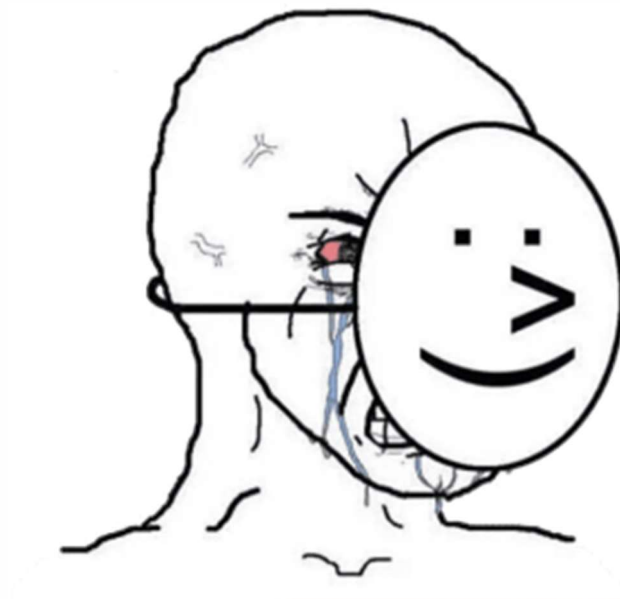
Miércoles 16/11:

- Corrección de detalles en el diagrama UML y agregación de notas.
- Corrección de detalles en la segunda versión del código.



Lunes 21/11:

[Feriado]



Miércoles 23/11:

- Aquí presentes tratando de sobrevivir T_T. Sean amables

