

Trabajo Practico Anual Entregables

Entrega 3

Materia: Diseño de Sistemas

Curso: K3002

Grupo: 4

Integrantes:

Chiara Beretta - Guadalupe García Fariñas,

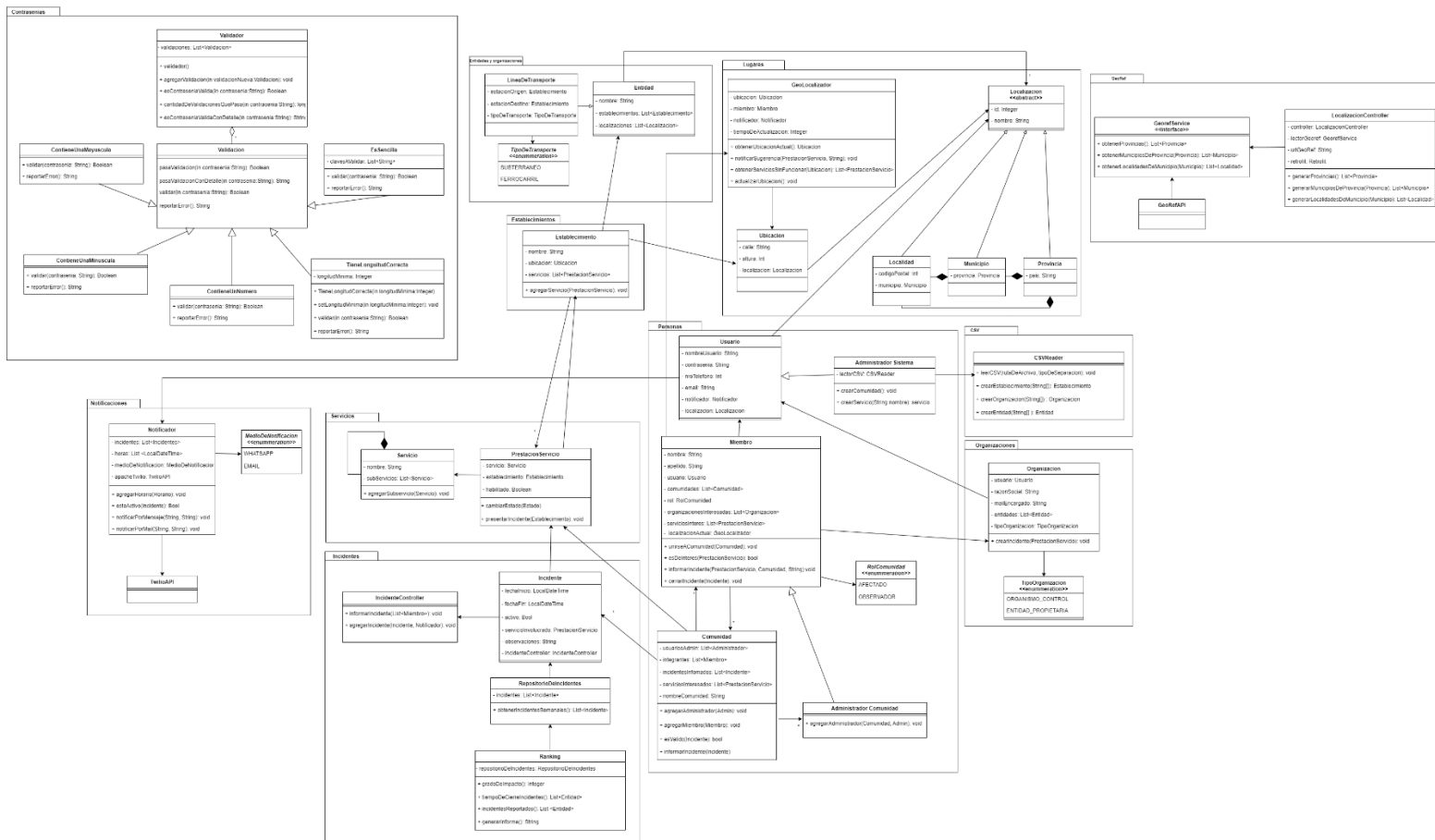
Ignacio Piloni - Nicolas De Rosso

Marcos Gurruchaga - Mora Rodriguez

Profesor: Luciano Straccia

Año:2023

Diagrama de Clases

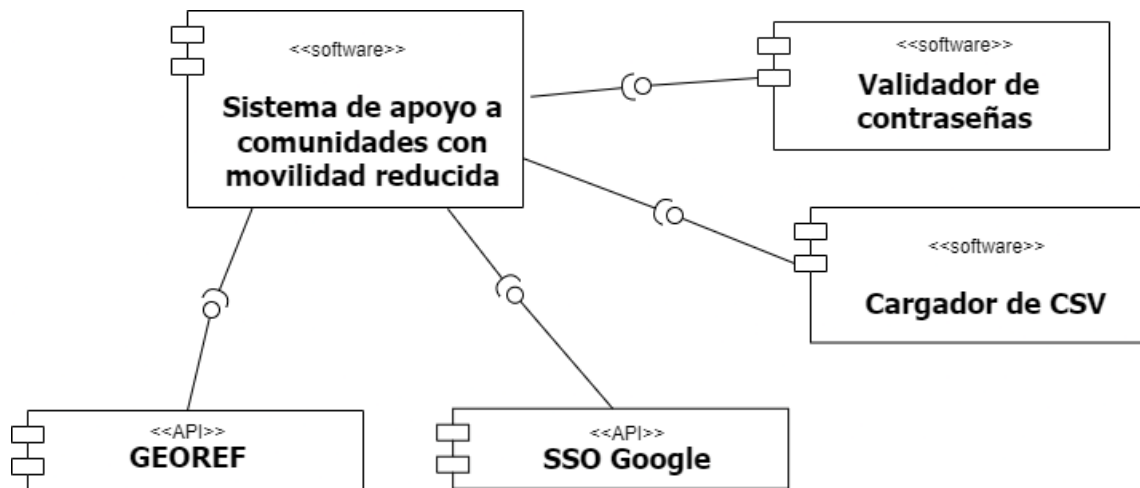


Link al repositorio con el diagrama: <https://github.com/ipiloni/DDS-TPA-Grupo4/tree/main/Entrega%203/Entregables>

Diagrama de Casos de Uso



Diagrama de Componentes



Decisiones de diseño

Ámbito de decisión (Objetos /Arquitectura / Persistencia / Otro)	Componente/s impactado/s	Decisión	Otras Alternativas	Justificación de la decisión
Objetos (Diagrama de clase)	Modelado de Servicios Públicos	Modelar una clase LineaDeTransporte, la cual contenga una lista de Estaciones	Modelar una clase Ferrocarril y una clase Subterráneo	Se trata a los ferrocarriles y subterráneos como la misma clase, lo cual permite polimorfismo y evita repetición de código.
Objetos (Diagrama de clase)	Modelado de Ubicaciones	Modelar una clase Ubicacion con sus respectivos atributos	Modelar Ubicacion como un String	Permite reutilizar a una instancia de la clase Ubicacion para estaciones que compartan la misma ubicación real.
Objetos (Caso de uso)	Modelado de actores	Modelar solo al actor "Miembro"	Modelar al actor "Miembro" y al actor "Persona"	Elegimos solo modelar al actor miembro porque hacía las mismas acciones que la persona sin comunidad y era redundante hacer un actor extra.
Objetos (Diagrama de clase)	Modelado de miembro y administrador	Modelar a la clase miembro de la cual hereda administrador	Modelar a la clase persona y dentro de la comunidad que figure dentro de la lista de un administrador	Para que un administrador pueda crear comunidades y tenga los métodos para gestionirlas, agregar nuevos admins, servicios, etc
Objetos (Diagrama de clase)	Modelado de Servicios	Modelar una clase Servicio con una lista de subservicios (de tipo Servicio) y una clase PrestacionServicio por cada servicio que se preste en una Estacion	Modelar una clase abstracta servicio la cual hereden ServicioSimple y ServicioCompuesto, con un método en común que sea estaDisponible(): Boolean	Modelar una clase abstracta nos obligaría a hacer una clase Servicio por cada Estacion que exista, y eso nos evita crear la clase PrestacionServicio; en cambio al modelar una clase Servicio la tratamos de forma abstracta para cualquier estación, y una clase PrestacionServicio por cada servicio en una estación en particular.

Objetos (Diagrama de clase)	Modelado de Prestacion Servicios	Que el estado sea un booleano "habilitado"	Agregar un enumeration estado como atributo dentro de la clase PrestacionServicio	Para el sistema solo importa si el servicio está o no en funcionamiento/habilitado
Objetos (Diagrama de clase)	Clase Localización	Se decidió crear una clase abstracta para la localización	Usar atributos nulos para el país, departamento o municipio cuando no se necesite	Mejor encapsulamiento y coherencia en el diseño: Al crear una clase abstracta para la localización, estás creando una entidad separada y específica para representar y manejar la información relacionada con la ubicación. Esto permite encapsular la lógica y los datos relacionados en un solo lugar, lo que resulta en un diseño más coherente y fácil de mantener.
Objetos (Diagrama de clase)	Clase Entidad	Usar a la clase entidad como clase	Usar a la clase entidad como clase abstracta y crear clases que hereden de ella para las distintas organizaciones	Como por ahora las distintas organizaciones no tienen distinto comportamiento, (excepto LineaDeTransporte) lo mantenemos como clase
Objetos (Diagrama de clase)	Clase LineaDeTransporte	Usar a la clase LineaDeTransporte como herencia de Entidad	Agregar los atributos de LineaDeTransporte a Entidad y utilizar la clase Entidad para cualquier entidad específica	Permite la reutilización de los atributos que tendrán todas las Entidades; mejora la modularidad debido a que se estructura más el código; permite que sea extensible dado que cualquier Entidad que tenga atributos propios serán herencia de la clase
Objetos (Diagrama de clase)	Clase Prestacion Servicio	Mantener la localizacion en el establecimiento	Agregar la localización a la prestación del servicio	La localización del servicio se puede saber preguntando en que establecimiento se encuentra la prestación y no hace falta agregar el atributo dentro de cada prestación
Objetos (Diagrama de clase)	Clase Establecimiento	Crear la clase establecimiento y que de ella se creen las instancias de estacion y sucursal	Crear la clase estacion que herede de establecimiento	Nos dimos cuenta de que, por ahora, la estación no tiene ningún comportamiento especial para la cual deba heredar de establecimiento, es decir, no hace falta diferenciarla en cuanto a atributos a la estación de un establecimiento cualquiera

Objetos (Diagrama de clase)	Clase Organizacion	Crear una clase Organizacion que sirva tanto para empresas, como para organismos de control y entidades propietarias	Crear una clase por cada organización (clase empresa, clase organismoDeControl)	Al no tener (por ahora) distinto comportamiento, optamos por generalizar a Organizacion
Objetos (Caso de uso)	Caso de uso "Visualizar informe de servicios"	El encargado ve el informe a través del mail.	Que pueda visualizarlo en el sistema	Esto les permite revisar los informes incluso cuando no tienen acceso directo al sistema.
Objetos (Diagrama de clase)	Clase Miembro	Crear un enum con el rol (observador-afectado)	Crear clases para cada uno que hereden de miembro	Por ahora no tienen comportamiento distinto y, por sobre todo, este rol tiene que poder ser modificado.
Objetos (Diagrama de clase)	Clase Incidente	Decidimos que al crear un incidente, se le agregue una observación (un String).	Agregar "observaciones" como una lista de strings.	Entendemos que el usuario solo puede hacer observaciones al momento de registrar el incidente, por lo cual no se agregarían más después.
Objetos (Diagrama de clase)	Clase AdministradorSistema	Decidimos crear esta clase y que herede de Usuario	Un booleano en usuario que diga si es administrador o no.	Un administrador de sistema no forma parte de las comunidades, por lo que tiene que ver con un usuario y no con un miembro; buscamos separar visiblemente el comportamiento de este usuario admin. del de un usuario común.