

Materia: Diseño de Sistemas

Código de Materia: 082028

Curso: K-3054

Docentes:

Mur, Pablo
Oliva, Miguel
Procopio, Demian
Rico Mendoza, René
Sosa, Ezequiel
Valido, Leandro

Trabajo Práctico: *"Nombre del trabajo práctico"*

Tipo: Grupal / Individual

GRUPO N° XX	
NOMBRE Y APELLIDO	LEGAJO N°
Salomone, Cecilia	112280-0
Jorge, Martinez Rodriguez	140831-8
Francisco Enzo, DiGiorgio	149744-3
Walter Hernan, Aguilar	127938-5
Javier, Salvatella	111320-3

Fecha prevista de entrega: 19/ 05/ 2016

Fecha real de entrega: 19/ 05/ 2016

Calificación..... Firma.....

Diseño de Sistemas	Curso: K-3054 – Año 2016
Trabajo Práctico: Entrega 2.5	Grupo: 11 – Versión 1.0

Historia de revisión

Fecha	Descripción	Autor	Versión
__/__/201X	-	-	1.0

Diseño de Sistemas	Curso: K-3054 – Año 2016
Trabajo Práctico: Entrega 2.5	Grupo: 11 – Versión 1.0

Tabla de Contenidos

1 Enunciado.....	3
2 Desarrollo.....	4

1 Enunciado

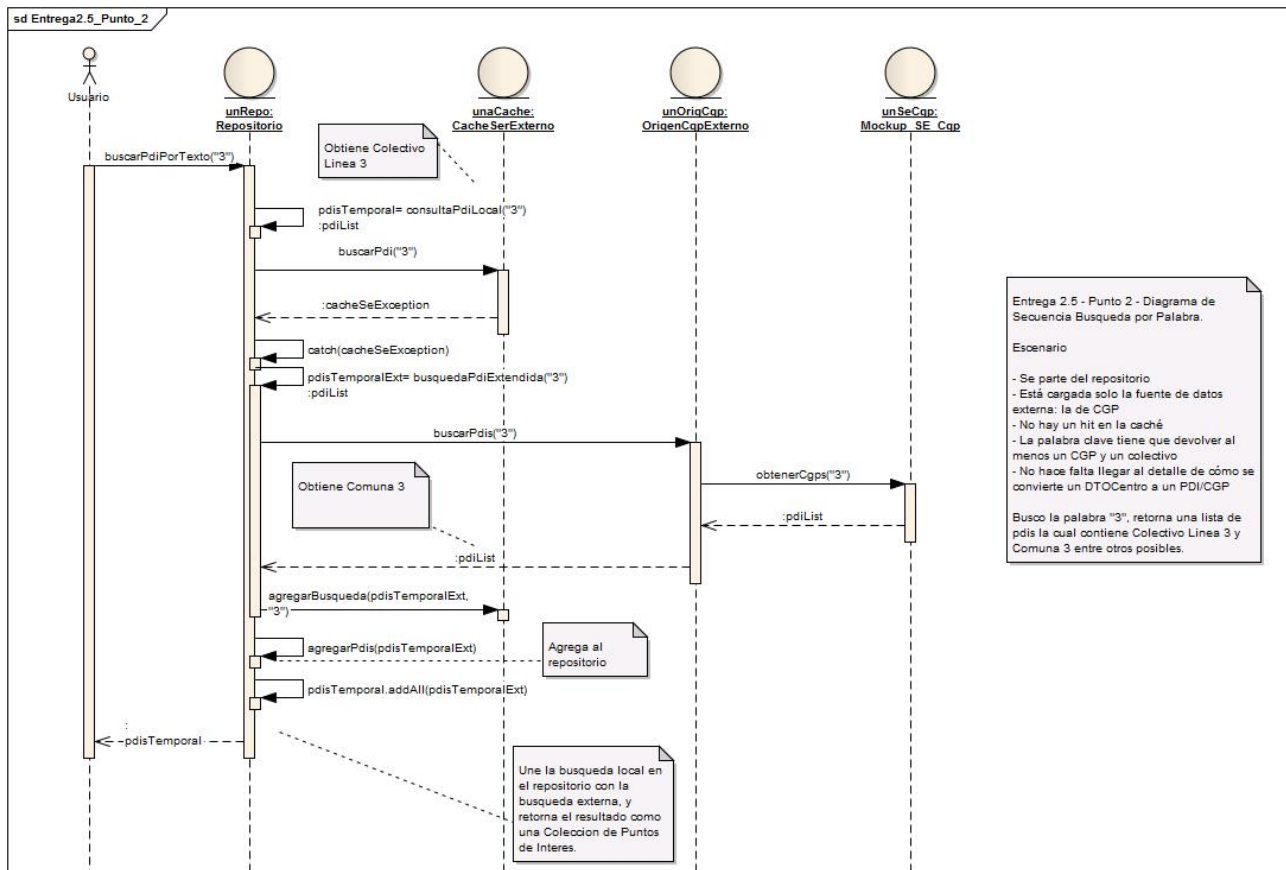
Documentación

Hay que crear un documento que posea los siguientes elementos:

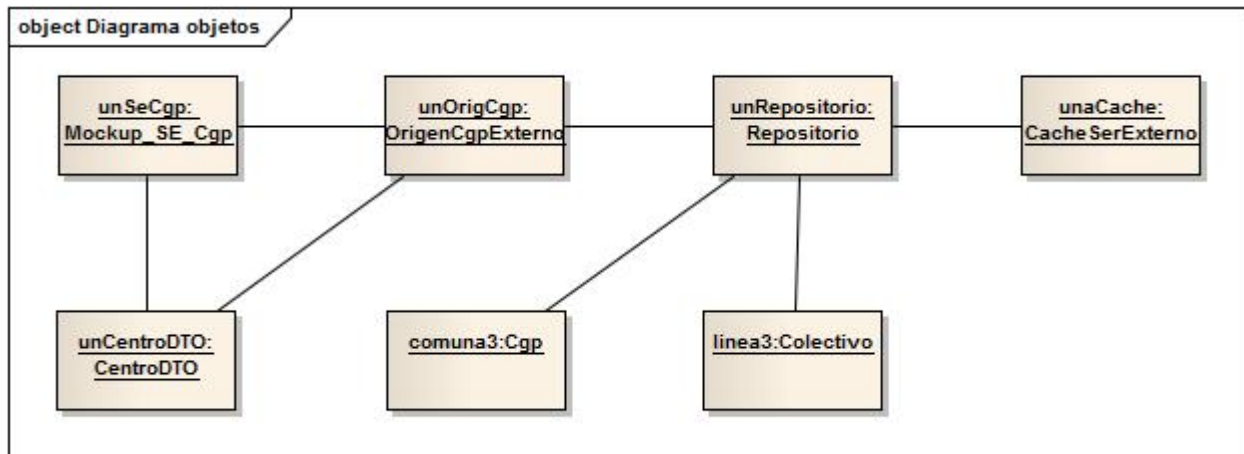
1. Diagrama de clases de la solución.
2. Diagrama de secuencia de una búsqueda por palabra donde:
 - a. Se parte del repositorio
 - b. Está cargada solo la fuente de datos externa: la de CGP
 - c. No hay un hit en la caché
 - d. La palabra clave tiene que devolver al menos un CGP y un colectivo
 - e. No hace falta llegar al detalle de cómo se convierte un DTOCentro a un PDI/CGP
3. Diagrama de objetos de la situación anterior.
4. Diagrama de colaboración de una búsqueda igual a la anterior solo que hay un hit en la caché.
5. Decisiones de diseño: la idea es que aquí se pongan en una lista de ítems, las discusiones que fueron teniendo con sus ayudantes.

Por ejemplo: modelaron parada? hay alguna entidad hayan querido modelar pero al final no lo hicieron? qué interfaces crearon, por qué no son clases abstractas? Usaron algún patrón o pensaron en usar pero luego decidieron que no era apropiado? Tienen que enumerar al menos cinco decisiones.

2. Diagrama de secuencia

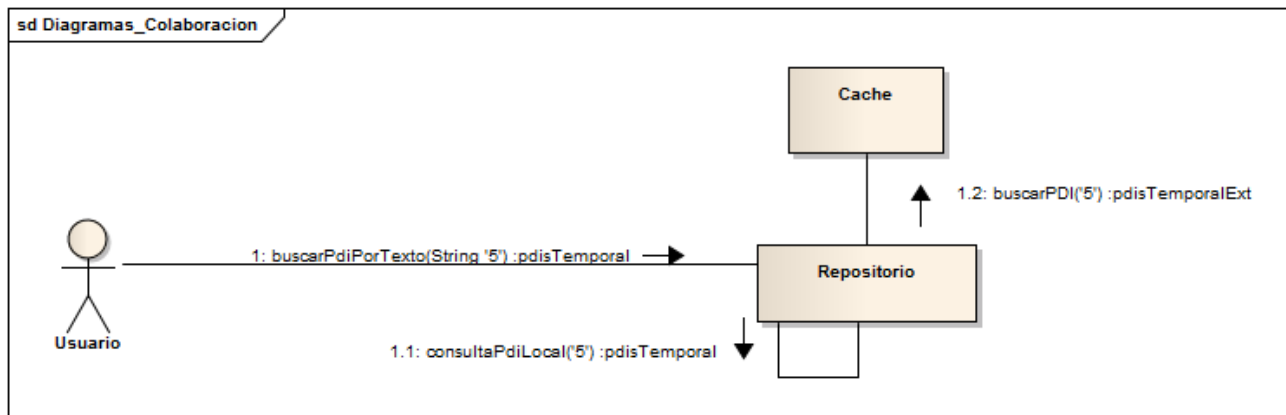


3. Diagrama de objetos



Diseño de Sistemas	Curso: K-3054 – Año 2016
Trabajo Práctico: Entrega 2.5	Grupo: 11 – Versión 1.0

4. Diagrama de colaboración



5. Decisiones de diseño

- Se utilizó una clase abstracta llamada PuntoInteres donde se agrupa toda la lógica de los puntos de interés, CGP, SucursalBanco, Colectivo, LocalComercial. Se definieron en ella métodos abstractos que implementan de forma independiente cada clase que está heredando. Aquí usamos el patrón template method.
- La parada no se modeló porque en su lugar se modeló una lista de puntos (POINT), ya que la parada es eso un punto, coordenada x e y.
- En el caso de los rangos horarios, se modeló la clase "RangoHorario" la cual contiene una lista de horarios que son una tupla 3 de elementos: día, hora de inicio y hora de fin; y 2 métodos los cuales son: agregar y pertenece, agregar agrega el horario a la lista de horarios y pertenece comprueba si la fecha introducida está dentro de algún horario.
- Se planteó la posibilidad de que horario sea una clase pero como no tiene métodos se optó por una tupla.
- Los servicios de CGPs y Bancos se acoplaron en una sola clase para separar comportamiento y delegar la búsqueda. De la misma manera los rangos horarios de los servicios y de los locales comerciales se unifican en una clase para separar comportamiento.
- Sobre la comunicación con los servicios externos se conversó con el ayudante, para encontrar una forma de hacerlo, pues anteriormente se había hecho que el Repositorio conociera a los servicios externos, que al ser diferentes, no se podían agrupar en una lista, lo que daba inflexibilidad para agregar nuevos servicios externos y tratarlos a todos igual.
- Las interfaces que usamos las hicimos con la intención de lograr un acuerdo en base a un método para distintas clases. Esto se utilizó para los orígenes externos, ya que el Repositorio tiene una lista de orígenes a los que le pido pdis y queríamos un método polimórfico que entiendan todos estos objetos. En este caso estamos usando el patrón Strategy y la estrategia es que busque pdis en PDIEexterno, en cada origen de datos se va a buscar de manera diferente. Y el repositorio se desentiende de como lo hace. No son clases abstractas porque no tienen atributos; sino que solo es la firma del método.
- De igual forma teníamos una Clase convertidor que luego de conversar con el ayudante nos dimos cuenta, con su ayuda que debíamos separarla en diferentes convertidores según sea el servicio externo que se trate.
- Se desacopló un poco el repositorio y el administrador que anteriormente tenían muchas responsabilidades. Se delegó en otras clases este comportamiento.
- En el caso de la búsqueda a servicios externos como el enunciado aclara de que uno es costoso y el otro es lento, decidimos modelar la clase CacheSerExterno la cual almacena búsquedas que se hicieron al servicio externo de manera de cuando se tenga que consultar al servicio externo, se consulte primero a la caché a ver si no se ha consultado antes, de ser así retorna lo que está en la caché, de lo contrario recién ahí busca en el servicio externo.