

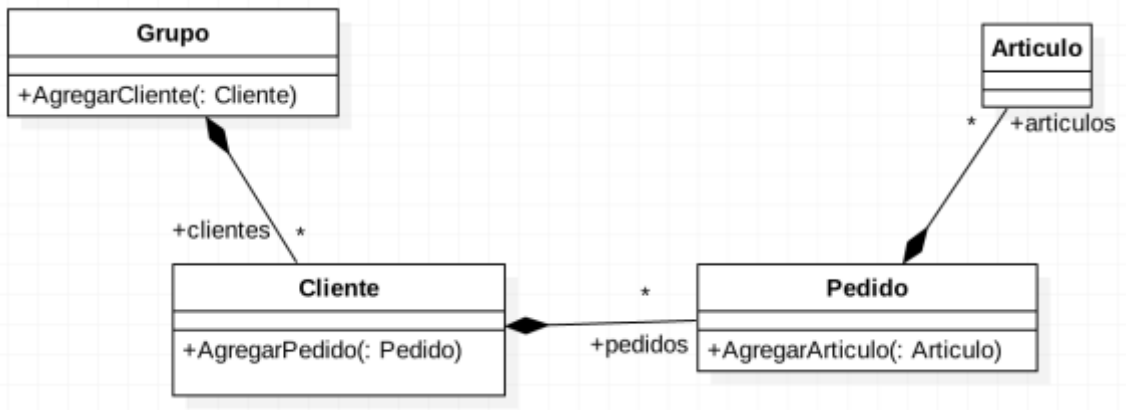
Importante:

- Escribir con letra clara.
- Todos los diagramas deben ser completos, prolijos y claros, incluyendo todos aquellos adornos UML, para asegurar la correctitud y no ambigüedad.
- La ausencia de cualquiera de los puntos anteriores puede implicar la quita de puntos.

Preguntas de Diseño (75 puntos)

1. Patrones y principios de diseño (35 puntos)

- a) (10 puntos) En el siguiente diagrama de clases se representa un *Grupo* de clientes, en el cual cada *Cliente* cuenta con uno o mas pedidos, y cada *Pedido* tiene uno o mas *Articulos*.



- (5 puntos) Se requiere crear un nuevo módulo de *Informes* en nuestra solución para poder realizar reportes sobre un grupo de clientes. Estos reportes necesitan analizar todos los datos relacionados con el cliente, por lo que todas las entidades involucradas en el diagrama Cliente, Pedido y Articulo deben poder ser visitados. Plantee mediante un diagrama de clases una solución a este problema aplicando el patrón *Visitor*.
 - (5 puntos) Realice un diagrama de interacción para su solución de la parte (i).
- b) (15 puntos)

Para el mantenimiento de una *Agenda*, se tiene información como los números de teléfono y direcciones, por lo que se cuenta con clases para el manejo de la *Direccion* y el *Telefono*.

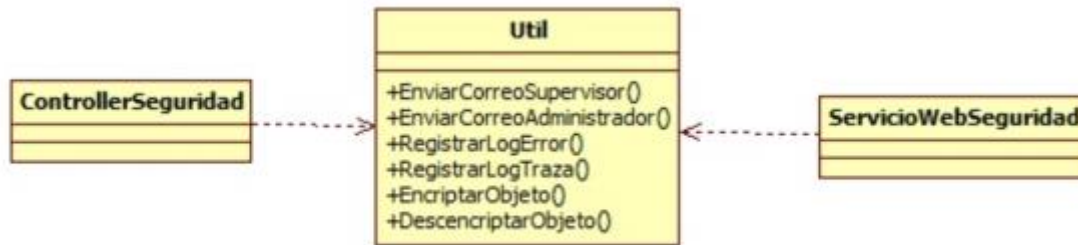
Tanto los números de teléfonos como las direcciones, tienen un comportamiento particular para cada país, por lo que se quiere tener diferentes implementaciones según el país de los datos a almacenar en la agenda.

Se quiere poder agregar nuevas implementaciones para los diferentes países con el menor impacto posible en la solución, agregando una nueva entrada en la agenda manteniendo la coherencia entre la dirección y teléfono perteneciente a un mismo país.

- i. **(5 puntos)** ¿Qué patrón de diseño puede utilizar para modelar esta situación? Justifique su respuesta.
- ii. **(5 puntos)** Dibuje un diagrama de clases lo más completo posible para modelar su solución de la parte (i).
- iii. **(5 puntos)** Muestre mediante un diagrama de interacción como se agregaría una nueva entrada a la agenda para un teléfono y dirección de Uruguay.

c) (10 puntos)

Dado el siguiente diagrama:



¿Qué puede decir según los SOLID? Justifique su respuesta y realice un nuevo diagrama de clases que mejore la solución de diseño.

2. UML (10 puntos)

Dada una aplicación distribuida, con arquitectura cliente - servidor, y definida por los siguientes componentes:

Nombre	Tipo	Descripción	Deploy en
Administracion	Ejecutable	Contiene la interfaz de usuario que utilizan los administradores del sistema. Utiliza clases definidas en Utilitarios.	PC del administrador
Administracion.exe.config	Archivo de configuración	Administracion.exe lee la configuración (por ejemplo ip y puerto servidor) desde este archivo.	PC del administrador
Servicios	Web API	Servicios REST que reciben los pedidos y los delega a las clases en la librería de clases LogicaNegocio.	Servidor de aplicaciones
Servicios.config	Archivo de configuración	Servicios lee la configuración (por ejemplo ip, puerto, conexión base de datos) desde este archivo.	Servidor de aplicaciones

Escuela de Ingeniería

Examen de: Diseño de Aplicaciones 2 (PLAN 2013)

Código de materia:

Fecha: 06-Feb-2017

Hoja 3 de 6

LogicaNegocio	Biblioteca de clases	Implementa toda la lógica de negocios, y utiliza clases definidas en Utilitarios y las interfaces en Auditoria.Interfaces.	Servidor de aplicaciones
Utilitarios	Biblioteca de clases	Clases para funciones comunes a varios módulos.	PC del administrador y servidor de aplicaciones
Auditoria.Interfaces	Biblioteca de clases	Define las interfaces con el comportamiento genérico de auditoría.	Servidor de aplicaciones
Auditoria.Implementacion	Biblioteca de clases	Implementación concreta de clases para la auditoria. La interface (ILog) está definida en Auditoria.Interfaces. Se utiliza mediante inyección de dependencias.	Servidor de aplicaciones
DB	Base de datos	Base de datos relacional en un servidor SQL Server.	Servidor de base de datos

a) (5 puntos)

Construya un diagrama de componentes que muestre, lo más detallado que pueda, la descripción anterior.

b) (5 puntos)

Muestre dichos componentes en un diagrama de entrega que incluya toda la información que pueda identificar.

3. Frameworks (10 puntos)

a) (5 puntos)

Cuando se utilizan en conjunto con patrones, bibliotecas de clases y componentes, los Frameworks pueden aumentar significativamente la calidad del software y reducir el esfuerzo de desarrollo, sin embargo, ¿qué desafíos se deben considerar para implementar y utilizar un Framework de forma efectiva?

b) (5 puntos)

Mencione dos beneficios / diferencias de los Frameworks respecto a las bibliotecas de clases.

4. Métricas de diseño (20 puntos)

a) **(15 puntos)** Realice un ejemplo concreto (utilizando diagramas, código, valores puntuales de las métricas) que demuestre la utilidad de tener los valores sobre el diseño de las siguientes métricas de diseño:

- Estabilidad (I)
- Abstracción (A)

b) **(5 puntos)**

¿Qué debería hacer con un componente con cohesión relacional (H) muy baja?

Preguntas de Tecnología (25 puntos)

(RESPONDER EN HOJA APARTE)

5. Componentes de Angular (10 puntos)

a) (5 puntos)

Explique en sus palabras el mecanismo de inyección de dependencias en el framework AngularJS, indicando una ventaja y una desventaja a la hora de usarlo en producción.

b) (5 puntos)

¿Para qué sirven los servicios en AngularJS? De una definición para fundamentar su respuesta y ejemplos de servicios ya incorporados en AngularJS.

6. Web API (15 puntos)

En vista de los cambios en las condiciones climáticas de nuestro país, el servicio de Meteorología desea crear una API que permita a los servicios de noticias y cualquier ciudadano consultar en tiempo real las predicciones del tiempo.

La api deberá retornar un objeto complejo Reporte, que se compondrá de la siguiente manera.

```
1  [
2    {
3      "hourlyState": [
4        {
5          "date": "2017-02-06T02:28:21+0000",
6          "humidity": 78,
7          "temperature": 24,
8          "state": "Nublado"
9        },
10       {
11         "date": "2017-02-06T03:28:21+0000",
12         "humidity": 84,
13         "temperature": 22,
14         "state": "Nublado"
15       }
16     ],
17     "currentState": {
18       "date": "2017-02-06T01:28:21+0000",
19       "humidity": 78,
20       "temperature": 24,
21       "state": "Nublado"
22     },
23     "specialAlerts": [
24       {
25         "start": "2017-02-06T07:28:21+0000",
26         "finish": "2017-02-06T17:28:21+0000",
27         "reason": "vientos",
28         "color": "yellow"
29       }
30     ]
31   }
32 ]
```

a) (10 puntos)

Se deberá desarrollar un endpoint que permita obtener dicho reporte, en la url '/api/reports', enviando por parámetros de url el departamento, ciudad y cantidad de horas para previsión futura. (solo debe desarrollar la clase del controlador).

Asuma que es dado una interfaz IWeatherService con un método GenerateReport, que recibe por parámetros los mismos que han sido enviados y retorna el reporte generado, cuya dependencia puede inyectarse al controlador.

Además, cuenta con las siguientes clases:

```
public class Report
{
    public State CurrentState { get; set; }
    public List<State> HourlyState { get; set; }
    public List<Alert> SpecialAlerts { get; set; }
}
```

```
public class State
{
    public int Temperature { get; set; }
    public int Humidity { get; set; }
    public String State { get; set; }
}
```

```
public class Alert
{
    public String Color { get; set; }
    public String Reason { get; set; }
    public DateTime Start { get; set; }
    public DateTime Finish { get; set; }
}
```

b) (5 puntos)

Explique en sus palabras las ventajas con las que cuenta a la hora de desarrollar dicho endpoint que generan que sea tan sencillo desarrollarlo (mínimo 3).