

 UNIVERSIDAD Popular del Cesar	UNIVERSIDAD POPULAR DEL CESAR PROGRAMACIÓN III PARCIAL 1	2021-II
--	---	----------------

Se solicita un software para Liquidar el hospedaje de los huéspedes en el hotel **Sol y Luna**. Para esto se hace necesario conocer: el tipo de habitación, la cantidad de personas a hospedarse, la fecha de ingreso, la fecha de salida y el tipo de huésped que podrá ser particular o socio.

- La liquidación del hospedaje para particulares

Está dada por la siguiente tabla:

Tipo	Cupos Permitidos	Valor Hospedaje Por día
Habitación Familiar	Hasta 4 personas	\$2000
Habitación Sencilla	1 persona	\$4000
Habitación Doble	2 personas	\$6000
Suite	Hasta 6 personas	\$12000

Para hospedajes entre 2 a 3 días se cobrará un adicional por día equivalente al 10% de la habitación.

A partir del 4 día se cobrará un adicional por día equivalente al 20% del valor de la Habitación.

- La liquidación del hospedaje para socios

Los socios tendrán derecho a tres días sin cobro en la suite del hotel para dos personas. Si las dos personas supera los tres días se cobrará un adicional por día equivalente al 5% del valor real de la suite.

Si requieren adicionar más personas en la suite el valor a liquidar estará dado por la siguiente tabla:

Cantidad de personas adicionales	Valor Hospedaje Por día
1	\$100
Más de 1 y menos de 4	\$200
De 4 a 6*	\$300

*No se podrá hospedar a más de 6 personas en la suite

Si el socio requiere liquidar un tipo de habitación diferente a la suite, se cobrará el 2% del valor de la habitación por cada día de hospedaje.

 UNIVERSIDAD Popular del Cesar	UNIVERSIDAD POPULAR DEL CESAR PROGRAMACIÓN III PARCIAL 1	2021-II
--	---	----------------

REQUISITOS DE LA SOLUCIÓN.

- Para la solución propuesta deberá aplicar la arquitectura en capas, las buenas prácticas de programación, los principios de programación.
- Para la gestión de los datos se utilizará un archivo plano, la forma de lectura y escritura en archivo es decisión del estudiante, podrá utilizar archivos secuenciales, binarios, XML, JSON, etc.
- Para la capa de Presentación se utilizará una aplicación de consola.
- El programa debe compilar para que pueda ser revisado, no se aceptan correcciones en la revisión.
- La solución desarrollada deberá ser gestionada con la herramienta de control de versiones Github, el repositorio deberá ser privado y deberá enviar una invitación al usuario **Anyamiyeth**.
- El link del repositorio de Github creado para gestionar la solución debe ser publicado en el Aulaweb máximo 20 minutos después de haber iniciado el parcial.
- El plazo máximo para realizar commit será el publicado en el AulaWeb.
- Al terminar cada punto, confirmará con el mensaje de commit." Punto x Finalizado" donde x será el número de punto que finalizó
- Los puntos son inclusivos y acumulativos, es decir para desarrollar cada punto debe haber desarrollado completamente el anterior, la revisión llegará hasta el punto que se haya completado correctamente.
- Los servicios deberán realizar la correcta gestión de errores, enviando por mensajes las posibles excepciones que pudieran generarse.

 UNIVERSIDAD Popular del Cesar	UNIVERSIDAD POPULAR DEL CESAR PROGRAMACIÓN III PARCIAL 1	2021-II
--	---	----------------

RETOS DE LA SOLUCIÓN

1. **(Nota Acumulada 0.5)** Las capas: Presentación, Entidad (Entity), Lógica (Business Layer Logical - BLL), Datos (Data Access Layer-DAL) y agregue las referencias necesarias para que la aplicación pueda funcionar con la arquitectura en capas.

Confirmar a Github el punto con mensaje de confirmación “Se confirma punto 1”

2. **(Nota Acumulada 1.5)** En la capa Entidad realizar la liquidación de acuerdo a las condiciones dadas, con las propiedades/métodos que se requieran para gestionar la información, recuerde aplicar los Patrones GRASP vistos en clases para la solución propuesta. **Ojo debe instanciar ambas liquidaciones desde el programa.cs mostrando que realiza el cálculo correctamente. No debe haber creado las Clases Service o Repository hasta este punto.**

Confirmar a Github el punto con mensaje de confirmación “Se confirma punto 2”.

3. **(Nota Acumulada 2.0, Guardar)** Implementar las clases de tipo servicio y repositorio que permitan desde Presentación:
 - a) Registrar la información necesaria para la liquidación solicitando: tipo de huésped, tipo de habitación, fecha de ingreso, fecha de salida y cupos.
 - b) Validar los cupos permitidos por habitación.
 - c) Calcular el valor del hospedaje de acuerdo a los requerimientos
 - d) Generar un consecutivo automático para identificar las liquidaciones registradas, incrementando en uno por cada nueva liquidación.

Confirmar a Github el punto con mensaje de confirmación “Se confirma punto 3”

4. **(Nota Acumulada 2.5, Consultar)** Implementar las clases de tipo servicio y repositorio que permitan desde presentación visualizar todas las liquidaciones realizadas, listando número de liquidación, tipo de huésped, tipo de habitación, fecha de ingreso (tipo de dato DateTime), fecha de salida (tipo de dato DateTime), cupos, cantidad de días de hospedaje, valor de la liquidación. **Confirmar a Github el punto, con mensaje de confirmación “Se confirma punto 4”**

5. **(Nota Acumulada 3.0, Eliminar)** Implementar el servicio y repositorio que permitan desde Presentación eliminar una liquidación solicitando el número de liquidación. Presentar la lista de los datos antes y después de eliminar. **Confirmar a Github el punto, con mensaje de confirmación “Se confirma punto 5”**

6. **(Nota Acumulada 5.0)** Agregar un nuevo tipo de huésped de tipo socio premium respetando los patrones GRASP (bajo acoplamiento, alta cohesión, experto en Información, controlador, fabricación pura y polimorfismo). Para este tipo de huésped el valor de las habitaciones será igual al 1% del valor de la habitación por cada día de hospedaje. **Confirmar a Github el punto, con mensaje de confirmación “Se confirma punto 6”**

 UNIVERSIDAD Popular del Cesar	UNIVERSIDAD POPULAR DEL CESAR PROGRAMACIÓN III PARCIAL 1	2021-II
--	---	----------------

Ejemplo para guardar fecha en un archivo.

```
public void Guardar(Persona persona)
{
    FileStream origen = new FileStream(ruta, FileMode.Append);
    StreamWriter writer = new StreamWriter(origen);
    writer.WriteLine(persona.Id + ";" + persona.Nombre + ";" +
    persona.FechaNacimiento.ToString("dd/MM/yyyy") + ";" + persona.Direccion);
    writer.Close();
    origen.Close();
}
```

Ejemplo para consultar fecha de un archivo.

```
public List<Persona> Consultar()
{
    List<Persona> personas = new List<Persona>();
    FileStream origen = new FileStream(ruta, FileMode.OpenOrCreate);
    StreamReader reader = new StreamReader(origen);
    string linea=string.Empty;
    while ((linea = reader.ReadLine()) != null)
    {
        Persona persona = Mapear(linea);
        personas.Add(persona);
    }
    reader.Close();
    origen.Close();
    return personas;
}

private Persona Mapear(string linea)
{
    string[] datos = linea.Split(';');
    Persona perosna = new Persona();
    Persona.Id = datos[0];
    Perosna.Nombre = datos[1];
    Persona.FechaNacimiento = DateTime.Parse(datos[2]);
    Persona.Direccion = decimal.Parse(datos[3]);
    return perosna;
}
```

[Obtener el número de días entre dos fechas en C# \(kyocode.com\)](https://kyocode.com/csharp/obtener-el-numero-de-dias-entre-dos-fechas/)



UNIVERSIDAD POPULAR DEL CESAR
PROGRAMACIÓN III
PARCIAL 1

2021-II