
	Principios y Fundamentos de la Ingeniería del Software	
	Examen convocatoria de Junio 2015	2º Grado en Ing. Informática 17/6/2015

**QuePasApp, Simple. Personal. Mensajería en tiempo real.**

**Duración: 3h**



**Ilustración 1. Pantalla de chat grupal (izqda) y de lista de chats (dcha).**

## Visión del negocio

Se desea diseñar una aplicación móvil, que los usuarios se descarguen en forma de app y que implemente una aplicación de mensajería centralizada.

Para ello, reunidos con el cliente, se acuerda que las funcionalidades que aporte la aplicación sean las siguientes:

- **Lista de contactos.** Cada usuario tiene una lista con sus contactos (usuarios con los que puede chatear), de los que almacena alias, foto, teléfono y texto de estado. Aparece una estrella para marcarlo como favorito.
- **Favoritos.** Es una lista de los contactos que han sido marcados seleccionando la estrella.
- **Ver chats.** Se muestra al usuario una lista ordenada cronológicamente de todos sus chat (conversaciones), ya sean individuales (chat con un usuario), o grupales (chat con varios usuarios). En este escenario, al usuario se le ofrecen tres posibilidades:

- **Iniciar un nuevo chat,** para lo que deberá seleccionar en primer lugar un contacto. Si ya existiese un chat abierto, se muestra al usuario.
- **Crear un grupo.** En este caso, se pide al usuario que indique el asunto del grupo (un nombre, limitado a 25 caracteres, que identifique al grupo) y, opcionalmente, una foto. A continuación se seleccionan los participantes.

Si el usuario desea **ver un chat** en concreto, pulsa sobre él y se le mostrarán los mensajes enviados ordenados cronológicamente junto al nombre (nick) de su autor. Desde dentro de un chat se puede:

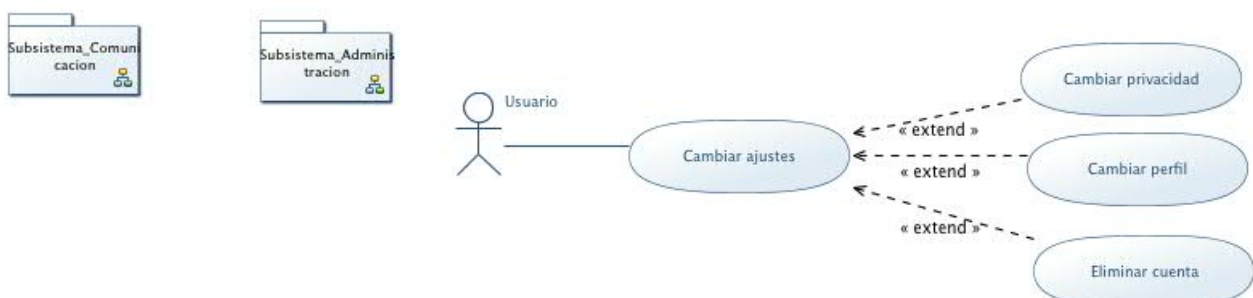
- **Ver información del chat:** En caso de ser con un contacto, vemos su número, foto y estado. En caso de tratarse de un grupo, se muestra nombre, foto, participantes (limitado a 100) y posibilidad de silenciarlo (a escoger de la siguiente enumeración: No, 8 horas, 1 semana, 1 año).
- **Escribir mensaje.** A la hora de comunicarnos con un usuario o grupo podemos compartir tres tipos de mensajes: **Texto**, una **imagen** (junto a un texto pie de foto) o una **ubicación**.
- **Borrar mensaje.** Podemos eliminar (de nuestro dispositivo) un mensaje concreto.
- **Reenviar mensaje.** Podemos reenviar cualquier tipo de mensaje a otro usuario, seleccionando primero el mensaje y después, seleccionando el contacto.
- **Cambiar ajustes.** Esta funcionalidad nos permite editar nuestro perfil (foto, alias y estado), editar la privacidad o eliminar la cuenta. En el apartado privacidad, se indica quién puede ver: la última vez que te conectaste, tu foto de perfil o tu estado. En los tres casos, se ha de escoger un valor de la siguiente enumeración (Todos, Mis contactos, Nadie)

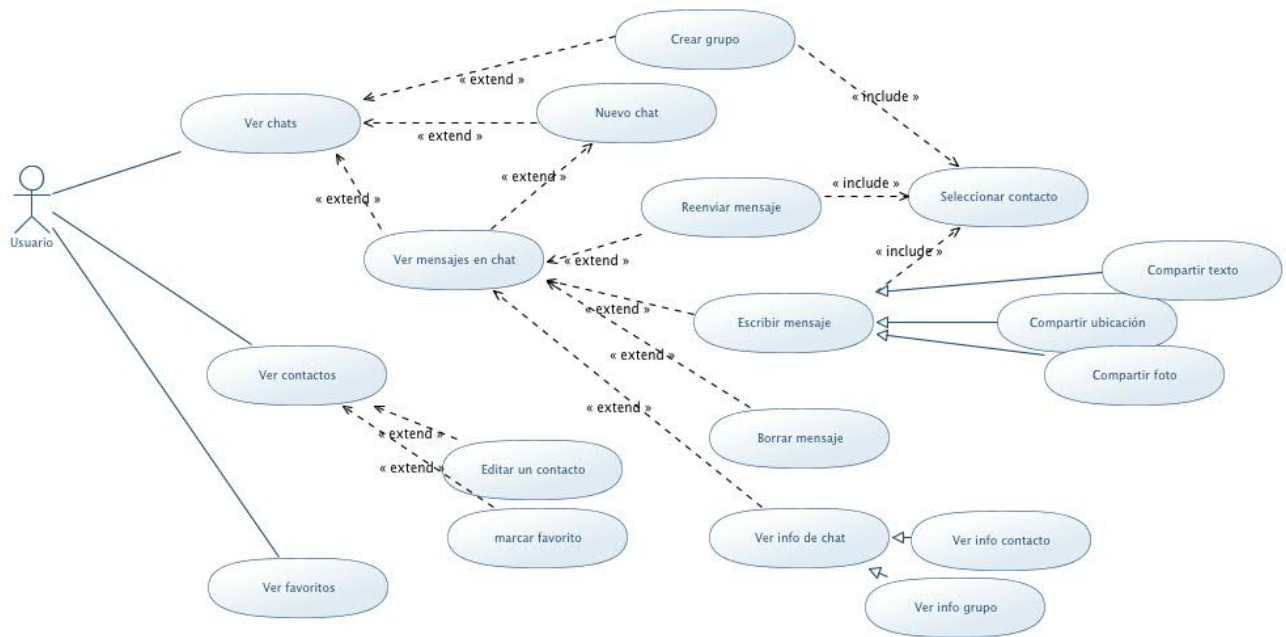
**Problema. Modelado (40% nota).** *Responda a este problema en folios continuos.*

Se le pide que analice, diseñe e implemente los siguientes entregables:

### 1. Diagrama de Casos de Uso

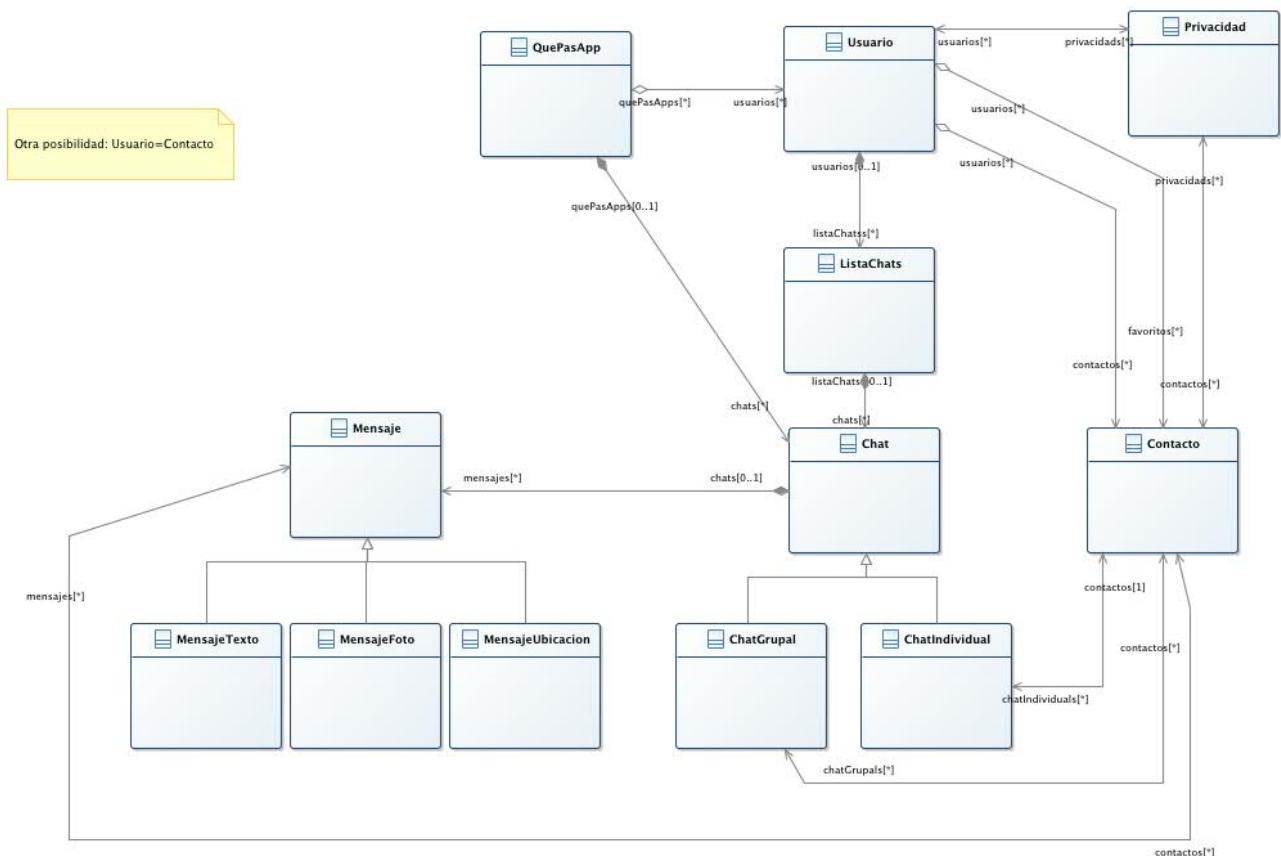
(2 puntos)





## 2. Diagrama de Clases (sólo atributos y relaciones)

(2 puntos)



## Pregunta. Los patrones (25% nota).

3. ¿Qué son los patrones de diseño? Defina el concepto, ponga un ejemplo, a ser posible, el que usó en la AAD y explíquelo brevemente. (1 punto)

Un patrón de diseño es una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias)...

El conocimiento de soluciones exitosas en el pasado permite al diseñador ser mas productivo y elaborar diseños mas flexibles y reutilizables.

Los patrones de diseño pretenden:

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.
- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

4. **¿Qué son los patrones GRASP?** Defina adecuadamente qué son y su utilidad.

(0,5 puntos)

*Patrones GRASP (Larman). General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). Describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones.*

*Los diagramas de interacción son algunos de los artefactos más importantes que se preparan en el ADOO. Es el momento de decidir que Clase se responsabilizará de realizar qué acciones para realizar los CU..*

*Cinco patrones GRASP:*

1. Experto en Información
2. Creador
3. Alta Cohesión
4. Bajo Acoplamiento
5. Controlador...

5. Indique para cada clase, qué realización de casos de uso le corresponde. (1 punto)

*Si bien existen diversas opciones, una posible solución sería:*

*ListaChats*

*Ver Chats*

*Chat*

*Ver mensajes en chat, Nuevo chat (abstracto), Crear grupo (sobrecarga de Nuevo chat)  
Nuevo chat individual (sobrecarga de Nuevo chat), Ver info de chat (abstracto)  
Ver info contacto (sobrecarga), Ver info grupo (sobrecarga)*

*Mensaje*

*Escribir mensaje (abstracto), Compartir texto (sobrecarga de Escribir mensaje)  
Compartir ubicación (sobrecarga de Escribir mensaje)  
Compartir foto (sobrecarga de Escribir mensaje)  
Reenviar mensaje, Borrar mensaje*

*Usuario*

*Seleccionar contacto  
Ver Contactos, Editar un contacto  
Cambiar ajustes, Cambiar perfil*

*Privacidad*

*Cambiar privacidad*

*QuePasApp*

*Eliminar cuenta*

**Problema. Estimación (15% nota).** Una vez identificadas las necesidades del cliente, llega el momento de estimar la duración y esfuerzo necesarios para realizar el proyecto. Se ha especificado la siguiente interfaz de usuario:

6. Basándose en la propuesta de interfaz, complete la siguiente tabla, para las siguientes transacciones:

- Enviar mensaje **de texto**
- Ficheros lógicos implicados en 'enviar mensaje de texto'

*Parece claro que se trata de una pantalla de entrada de datos, en este caso de mensajes, que se almacenan en el sistema y se despliegan a los destinatarios. Se nos pide que analicemos también a los ficheros lógicos implicados.*

(0,5 puntos)

Mirando el Diagrama de Clases podemos apreciar que la información necesaria para esta transacción se encontraría en las siguientes clases (posteriormente ficheros lógicos): el Chat concreto en el que se visualiza y el Mensaje en sí mismo.

Transacción	Tipo de Componente (EE, SE, GLDI, GLDIZ Y CE)	Número de ficheros y datos elementales	Lista de datos elementales
Enviar mensaje de texto	EE	Chat, Mensaje (Texto) F:2 DE: 1-4	Id_Chart, Id_Mensaje, texto
Chat	GLDI	1 tipo de registro	Id_Chart, nombre,... < 19
Mensaje (Texto)	GLDI	1 tipo de registro	Id_Mensaje, texto,... <19

## 7. ¿Calcule el total de puntos de función no ajustados (PFNA)?

(0.5 puntos)

$$FA=0,65+(0.01 \cdot SVA)$$

$$PFA=FA \cdot PFNA$$

DESCRIPCIÓN	SENCILLA	MEDIA	COMPLEJA	TOTAL P.F.
Nº de Entradas Externas	1 x 3	x 4	x 6	3
Nº de Salidas Externas	x 4	x 5	x 7	
Nº Grupos Lógicos de Datos Internos	2 x 7	x 10	x 15	14
Nº de Grupos Lógicos de Datos de Interfaz	x 5	x 7	x 10	
Nº de Consultas Externas	x 3	x 4	x 6	
TOTAL PUNTOS FUNCIÓN NO AJUSTADOS (PFNA)				17

A la hora de ajustar los PF, se ha de tener en cuenta que de los Valores de Ajuste, 1 vale 0, 3 valen 2, 4 valen 4 y los restantes valen 5.

14 factores: Restantes son 14-1-3-4=6;  $SVA=1 \cdot 0+3 \cdot 2+4 \cdot 4+6 \cdot 5=52$ ;  $FA=0,65+0,52=1,17$

$PF=1,17 \cdot 17=19,89PF$

## 8. ¿Si el proyecto empieza el próximo lunes, día 22 de junio, con 2 personas trabajando, cuándo se acabará?

(0.5 puntos)

Como ayuda se nos dice que se trabajan 8 horas diarias y 40 semanales. Se nos entrega la siguiente tabla, que recoge los valores obtenidos en los últimos proyectos similares:

Proyecto	PF	Personas	Horas
A	160	8	240
B	200	10	280
C	189	9	252
D	192	6	448

La mayoría de los métodos de estimación se basan en el análisis de características y datos de anteriores proyectos para intentar pronosticar cómo puede comportarse un nuevo proyecto.

Entre las técnicas usadas se encuentran fundamentalmente las estadísticas, como las técnicas de regresión lineal. En este caso, en que nos dan información básica sobre un número reducido de proyectos, intentaremos caracterizar nuestros proyectos con técnicas básicas, como la media.

Necesitamos conocer cómo se convierten los PF a Horas-persona. En vez de usar datos basados en tablas ajenas, como la del ISBSG, vamos a obtener la media de horas por punto-función en nuestros proyectos.

Según la tabla dada, obtenemos relaciones en cada proyecto y hacemos la media (horas/pers-PF):

A,  $240H \cdot 8personas=1920horasTotales$ ;  $1920h/160PF=12 H/PF$ ;

B,  $280H \cdot 10personas=2800horas$ ;  $2800/(200PF)=14 H/PF$

C,  $252H \cdot 9personas=2268horas$ ;  $2268/(189PF)=12 H/PF$ ;

D,  $448H \cdot 6personas=2688horas$ ;  $2688/(192PF)=14 H/PF$

Media conversion=13 H/(PF-pers)

Para este proyecto,  $19,89PF \cdot 13H/PF / 2pers = 129,28$  horas  $= 129,28/8h/día = 16,16$  días

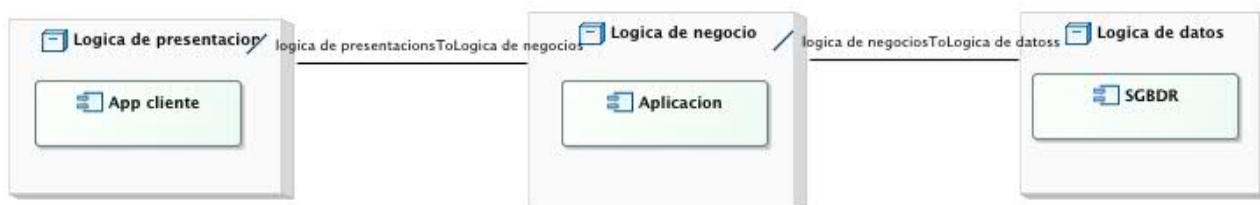
El proyecto empezará el 22 de junio y acabará 17 días laborables después, durante el 15 de julio

### Pregunta. Arquitectura del Sistema (10% nota).

Nuestros analistas consideran que la arquitectura más adecuada para nuestro sistema es una arquitectura cliente-servidor en tres capas. Recuerdo que la aplicación usa una app, el repositorio de mensajes/usuarios se encuentra centralizado y nos apoyamos en un sistema gestor de bases de datos relacional.

9. Realice el Diagrama de Despliegue correspondiente (con los componentes incluidos en los nodos).

(1 punto)



### Problema. Construcción (10% nota).

Una vez terminados el análisis y modelado, y realizada la estimación del proyecto, empezamos a codificar. Para construir el software, uno de los primeros pasos será, a partir del **Diagrama de Clases** obtenido, implementar en Java la estructura de las clases.

10. Realice la implementación estructural en Java de las clases **Mensaje y Usuario** (incluya sólo la declaración de los atributos, ni incluya ningún método, ni los *getters* y *setters* de los atributos).

(1 punto)

```
public abstract class Mensaje
{
    private Chat enChat;
    private Contacto autor;
    ...
}
```

```
public class Usuario
{
    private Privacidad miPrivacidad;
    private <ArrayList>Contactos misContactos;
    private <ArrayList>Contactos misFavoritos;
    private ListaChats misListasdeChats;
    ...
}
```