

# IIC 2143 Ingeniería de Software

## Interrogación 1 - Semestre 2 /2018

### Pregunta 1 (Relatos de Usuario)

Actores involucrados

- habitante de un depto
- responsable de un depto
- administrador del edificio
- administrador de la plataforma (aplicación)

Los relatos más importantes agrupados por temas

#### 1. Gastos Comunes

##### **Generar cobros de gastos comunes a cada departamento**

Como administrador del edificio necesito generar un detalle de lo que le corresponde pagar a cada departamento de modo que ellos puedan consultarlo antes de proceder al pago

##### **Obtener detalle del mes**

Como responsable de un depto necesito obtener lo que me corresponde pagar incluyendo el detalle de todo lo gastado para proceder a hacer el pago

##### **Pagar los gastos comunes**

Como responsable de un depto necesito poder hacer el pago de los gastos comunes de modo de no caer en categoría de moroso y exponerme a multas

##### **Conocer a los departamentos morosos**

Como habitante de la comunidad me interesa poder saber quienes no están pagando a tiempo los gastos comunes de modo de poner presión social a los infractores

#### 2. Ingreso de Gastos

##### **Ingresar gastos varios**

Como administrador del edificio necesito ingresar un nuevo gasto (monto y glosa) de modo de que pueda ser contabilizado en los cálculos a fin de mes

##### **Generar planilla de sueldos**

Como administrador del edificio necesito generar los montos que corresponde a los sueldos de cada uno de los empleados del edificio

#### 3. Gestión de espacios comunes

##### **Reservar un espacio común**

Como habitante del edificio necesito poder reservar alguno de los espacios comunes del edificio de modo de poder hacer uso de él en la fecha y hora indicada

##### **Cancelar una reserva**

Como habitante del edificio necesito poder cancelar una reserva de un espacio común de modo de dejarlo disponible para alguien mas

4. Informar a administrador

**Informar necesidad de reparación**

Como habitante del edificio necesito informar sobre algo que requiere ser reparado de modo de que el edificio se mantenga en impecables condiciones

**Informar problema de convivencia**

Como habitante del edificio necesito informar sobre algo que rompe reglas de comunidad del edificio de modo de poder mantener la convivencia de la comunidad

5. Socialización

**Conocer a vecinos**

Como habitante del edificio quisiera conocer los nombres y edades de quienes viven en un departamento dado para aumentar la cohesión de la comunidad

**Ofrecer un servicio**

Como habitante del edificio quisiera poder ofrecer un servicio (clases particulares, asesoría legal, baby sitting, cuidado de perros, etc) al resto de la comunidad para beneficio de todos

6. Plataforma

**Obtener estadísticas de uso**

Como administrador de la plataforma me interesa conocer para cada edificio que utiliza la plataforma el grado de utilización que hace de ella para poder ayudar o incentivar si es necesario

**Obtener edificios con pagos pendientes**

Como administrador de la plataforma necesito conocer los edificios que no se encuentran al día en sus pagos del servicio de la plataforma para inicialmente recordarles pagar y eventualmente cortar el servicio

Observaciones

- Pueden haber muchos mas relatos pero deberían poder haber identificado a lo menos unos 8 o 9 de estos.
- Es importante que estén escritos en la forma indicada
- No es necesario que todos tengan condición de satisfacción pero mínimo dos de ellos deben incluirla. Por ejemplo:

1.1 Generar cobros de gastos comunes

Verificar que para cada departamento se haya generado un cobro

Verificar que los cobros sean proporcionales a los m2 de cada departamento

Verificar que en el caso de deudores morosos se sume lo adeudado mas los intereses

1.3 Pagar los gastos del mes

Verificar que se pueda hacer por transferencia bancaria

Verificar que se pueda hacer por pago con tarjeta de crédito

Verificar que se pueda hacer por paypal

## **Pregunta 2 (Casos de Uso)**

### **a) Actores y Casos de Uso**

#### **Actores Primarios**

Ciclista activo

Administrador de la plataforma

#### **Actores Secundarios**

API de Tweeter

API de Facebook

API del Servicio Meteorológico

API de Google Maps

#### **Casos de Uso**

Registrar Nuevo Ciclista

Crear una Nueva Salida

Postear nueva salida en redes sociales

Inscribirse para una Salida

Consultar próximas salidas

Ingresar un tip en una categoría

Crear categoría de tips (lugares, equipos, nutrición, etc)

Comentar sobre una salida

Comentar sobre un tip

Consultar detalles sobre Salida

Obtener estadísticas de ciclistas

Obtener estadísticas de salidas

#### **Observaciones:**

- Pueden haber algunos otros actores primarios pero esos dos por lo menos tienen que estar
- Al menos debe haber un actor secundario. Pueden haber algo como "subsistema X" en lugar de API
- No es necesario que hayan mencionado actores fuera de escena
- Es importante el nombre de los casos de uso. Tienen que comenzar con un verbo activo en infinitivo (por ejemplo "Registro de un Ciclista" no estaría correcto) y tienen que dar una buena idea de la meta.

## **b) Caso de Uso: Inscribirse en una salida**

### Flujo Principal

1. Ciclista se autentifica en la aplicación
2. Ciclista elige seleccionar salida
3. Sistema despliega todas las próximas salidas de más próxima a más lejana
4. Ciclista seleccionar la salida que le interesa
5. Sistema despliega detalles de fecha, hora, lugar, número de participantes y pide confirmación
6. Ciclista confirma salida
7. Sistema le indica que ya ha sido registrado y le envía un correo con los detalles

### Flujos Alternativos

- 1a Ciclista no se ha registrado en la aplicación  
1a1 Sistema le indica que debe registrarse previamente  
1a2 Se ejecuta el Caso de Uso "Registrar nuevo Ciclista"

- 1b Ciclista no ingresó bien sus datos para autenticarse  
1b1 Sistema pide que corrija sus datos  
1b2 Ciclista ingresa nuevamente sus credenciales  
Se repite autenticación un máximo de 3 veces. Si no es exitosa el caso de uso termina.

- 3a. No hay salidas programadas  
3a1 Sistema indica que por el momento no hay salidas programadas  
3a2 El caso de uso termina

- 4a Al ciclista no le gusta ninguna de las salidas ofrecidas  
4a1 Ciclista selecciona salir de la aplicación

- 7a. La capacidad ya está completa por lo que no es posible agregarlo a la salida (puede ser que haya demorado demasiado en confirmar la salida y otros se inscribieron antes)  
7a1 El Sistema indica que lamentablemente no se le puede agregar a la salida indicada por capacidad.  
7a2 El Sistema le da la opción de seleccionar una nueva salida (paso 3)

- 7b. El ciclista no satisface las restricciones de edad por lo que no es posible agregarlo a la salida  
7b1 El Sistema indica que lamentablemente no se le puede agregar a la salida indicada por no satisfacer las restricciones de edad.  
7b2 El Sistema le da la opción de seleccionar una nueva salida (paso 3)

### Observaciones:

- No es muy importante que no esté el resto de los elementos (precondiciones, postcondiciones, alcance, etc)
- Es muy importante que el flujo principal corresponda SOLO al happy path y esté escrito en forma de diálogo entre el actor y el sistema
- Es muy importante que se identifiquen a los menos unos 4 flujos alternativos
- Es muy importante que cada flujo alternativo esté asociado al punto en el flujo en que se produce (por el número) y que se indica que es lo que ocasiona este flujo

### Pregunta 3 Scrum

a) Todos los métodos ágiles

- son iterativos e incrementales
- favorecen la colaboración con el cliente frente a la negociación de contratos
- favorecen las interacciones entre las personas por sobre los procesos y documentos
- priorizan el software por sobre la documentación

b) Dado lo mucho que tienen en común los métodos ágiles se pueden combinar prácticas. Por ejemplo:

- Se pueden hacer sprints y tener equipos multifuncionales (Scrum) pero programar en pares (XP)
- No hay problema en usar relatos de usuario o casos de uso tanto en Scrum como en XP
- El product owner es totalmente compatible con XP que pide que el cliente se haga parte en el desarrollo

c) El equipo de desarrollo Scrum es multidisciplinario por lo que los 3 desarrolladores deben ser usados para desarrollar la aplicación. Falta un scrum master y un product owner. Si hay alguno de los 3 que está certificado o tiene un training de scrummaster puede asumir ese rol. Si nadie tiene esa experiencia puede contratarse un consultor por horas para que apoye en esta función al equipo. El product owner DEBE venir de las áreas de negocio, alguien que entienda que es lo que le da valor a EDD y que por lo tanto pueda priorizar el backlog.

d) Sabemos que hay 3 desarrolladores y cada uno es capaz de implementar 4 requisitos funcionales por mes. Esto nos da una capacidad teórica de abordar 12 requisitos por mes. Si elegimos sprints de dos semanas de duración el número de requisitos a abordar es del orden de 6 por sprint, es decir necesitaremos unos 6 sprints de dos semanas para completar el proyecto.

Cuales funcionalidades irán en el primer sprint corresponde a las de mas alta prioridad en el backlog una vez que éste ha sido priorizado por el product owner.

Mas realísticamente es muy probable que 3 programadores no son capaces de hacer 3 veces lo que hace uno por mes (Brooks) por lo que 6 funcionalidades por sprint es muy optimista. Mas realísticamente los 3 desarrolladores podrían hacer 2 veces lo que hace 1, es decir unos 8 por mes o 9 sprints de 4 features cada vez.

#### **Pregunta 4 Aspectos Generales de IS**

a)

El producto no satisface al cliente (no es lo que el usuario necesitaba)

El proyecto se completó y recién ahí el cliente se da cuenta que hay problemas fundamentales que tienen que ver con una mala comprensión de los requisitos o que los requisitos cambiaron durante el proyecto

El desarrollo no se puede completar dentro de los plazos o del presupuesto acordado

El proyecto se atrasa significativamente (50 a 100% de tiempo adicional). Esto se traduce normalmente en un aumento importante en los costos (de desarrolladores)

b)

- El modelo de cascada es
  - muy simple de entender
  - funciona en otras áreas de la ingeniería
  - permite dibujar cartas gantt para el proyecto entero
  - permite separar tareas y equipos asociados
  - le da al jefe de proyecto un sentido de avance

c)

Que los interesados entienden perfectamente lo que necesitan

Que es posible extraer desde los interesados lo que realmente necesitan en un plazo relativamente corto (requisitos)

Que es posible expresar esos requisitos completamente en papel

Que las cosas (requisitos) no van a cambiar una vez que el proyecto comienza

d)

Que bastan documentos para traspasar información de una persona a otra (documentos largos formales)

Que los requisitos se pueden congelar una vez que el desarrollo comienza (la firma del cliente en el documento)