

Un proyecto de

Universidad de málaga

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

**REPOSITORIO:** [**https://github.com/chemafdezuma/softpro.git**](https://github.com/ChemaFdezUma/SoftPro.git)

**Álvaro Yuste Moreno**

**José Luis López Ruiz**

**Pablo Alarcón carrión**

**Adrián Camacho ferrezuelo**

**Marta González Palmero**



[**yuste@uma.es**](mailto:yuste@uma.es)

[**pepefx.lr@uma.es**](mailto:pepefx.lr@uma.es)

[**paproka@uma.es**](mailto:paproka@uma.es)

[**adricam@uma.es**](mailto:adricam@uma.es)

[**mgonzalezpalmero@uma.es**](mailto:mgonzalezpalmero@uma.es)

planificación

del

proyecto

**Álvaro valencia villalón**

**Rafael ceballos martínez**

**Rocío Gómez Mancebo**

**Jose María Fernández Cantón**

[**valeal@uma.es**](mailto:valeal@uma.es)

**rafacebamar@uma.es**

**rociogoomez27@uma.es**

[**josemariafcanton@uma.es**](mailto:josemariafcanton@uma.es)

Índice

**Introducción 2**

**Roles 2-3**

Aplicación 3

Uso de Scrum 3-4

**Gestión del riesgo 5**

**H**erramientas usadas **6**

**Requisitos 7-16**

**Casos de uso 16-26**

**Diagrama de Clases 27**

**Pruebas JUnit 28-29**

Introducción

Explicación del problema

Málaga es una ciudad que hoy en día recibe a miles de turistas, estudiantes, empresarios… Cada uno llega con un propósito específico, pero una vez aquí resulta inevitable querer conocer un poco más los sitios y oportunidades que puede ofrecer. Nuestro objetivo es llevar a cabo una aplicación que, a través de ciertas preguntas como número de personas para el plan, tipo de plan, dinero que se quiere gastar, el tiempo que hace, ubicación… Proponga al usuario el mejor plan que puede ofrecer la ciudad.

Roles

Del equipo

Los diferentes roles que se han asignado son, básicamente, 6:

* PO (Product Owner): se encargará principalmente de guiar el diseño del producto, con una meta clara y concisa. Representa a los usuarios y clientes del producto en cuestión
* SM (Scrum Master): permite que los miembros del equipo tengan las herramientas necesarias para trabajar correctamente, y modera las reuniones que tenga el equipo. Además, hace de coach y de administrador del proyecto.
* Equipo de desarrollo
  + Code (Programador): como su nombre indica, se encarga de la programación front-end y back-end del proyecto en cuestión, así como todo lo complementario a código que sea necesario.
  + Testing (Pruebas): se encarga principalmente de poner a prueba las diferentes versiones que tenga el producto, para así encontrar posibles bugs y fallas del mismo. Este rol es esencial dentro de la calidad a presentar del producto final.
  + Graphic Desing: Se encarga de diseñar los elementos gráficos (entre ellos, la presentación en diapositivas).
  + App Design: Su función es la organización, percepción visual y elementos adicionales que formarán parte de la percepción visual de la aplicación.
  + Risks (Posibles riesgos): El atributo que se le asigna se basa en buscar los riesgos principales que tenga cada fase del desarrollo, y notificar acerca de posibles soluciones a los mismos, en caso de que ocurriesen.
  + Analist (Analista): Identificar los requisitos que tendrá la aplicación coordinándose con el Product Owner.
  + Structural Designer: Idea y desarrolla los modelos necesarios para llevar a cabo el producto a bajo nivel.

Ya explicados, los miembros del grupo poseen los siguientes roles:

* Rafael Ceballos: Graphic Desing / Risks
* Álvaro Yuste: Structural Design / Analist
* Adrián Camacho: Code Specialist / Analist
* Pablo Alarcón: Code Specialist / Testing
* José M. Fernández: Code Specialist / Structural Design
* José Luis López: Code Specialist / App Design
* Rocío Gómez: Product Owner / Testing
* Marta González: Scrum Master / Structural Design
* Álvaro Valencia: Scrum Master / Graphic Design

Debido a la alta demanda de desarrolladores de código y teniendo en cuenta el equipo que tenemos, todos pertenecemos al departamento “code”. También dándonos cuenta nuestra inexperiencia diseñando y que cada uno tiene ciertas habilidades al respecto, hemos decididos fraccionar el departamento de “design” como viene reflejado arriba.

Icono

Descripción generada automáticamenteAPLICACIÓN

logo y nombre

El nombre ‘where2go’ se ha elegido teniendo en cuenta la funcionalidad principal de la aplicación, encontrar un plan acorde a tus necesidades.

El diseño del logotipo reúne todas las partes del nombre de la aplicación, en forma de símbolo abstracto y moderno.

'Where2go'

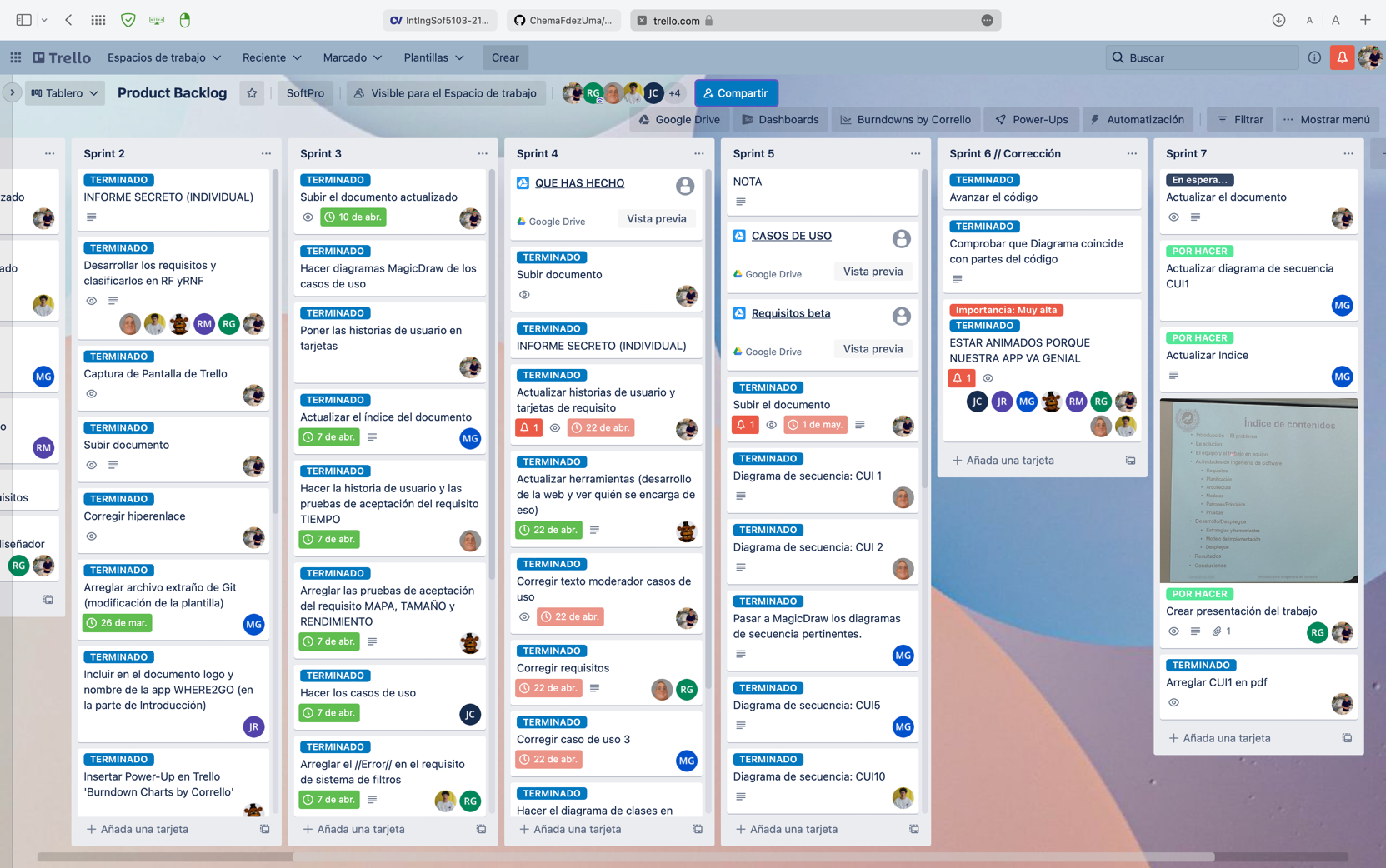
USO DE SCRUM

Y POR QUÉ LO HEMOS ELEGIDO

Creemos que este es el método más adecuado para este proyecto porque fomenta el trabajo en equipo, nos permite realizar cambios durante el proceso del proyecto y favorece el contacto con el cliente en todo momento.

Seguramente una vez terminado el proyecto habremos aprendido mucho sobre la metodología, las necesidades y habilidades de desarrollo de un proyecto y las capacidades necesarias para el trabajo en grupo, la colaboración y la planificación de actividades complejas.

Concretando, nuestra estrategia será la siguiente:

* Sprints de 1 semana de duración
* Sprint Planning (reunión grande) los lunes.
* Sprint Review Meeting (reunión de 15 minutos aproximadamente) los domingos

Gestión del riesgo

Tipo, probabilidad, efectos y mitigación

| Tipo | Riesgos | Probabilidad | Efectos | Mitigación |
| --- | --- | --- | --- | --- |
| Proyecto | Poca experiencia organizativa | Muy alta | Tolerable | Aprender lo máximo posible del profesor |
| Proyecto | Mala planificación del tiempo invertido en tareas | Alta | Serio | Tratar de evitar las máximas dependencias posibles y reducir el camino crítico en la planificación |
| Proyecto | Imposibilidad de implementación de las herramientas CASE | Alta | Tolerable | Tener varias herramientas disponibles por si una falla |
| Proyecto y producto | Cambio de diseño para implementar las funciones requeridas | Moderada | Serio | Optimizar lo máximo el código para que no suponga una pérdida de tiempo grande |
| Producto | La base de datos usada en el sistema no es capaz de procesar tantas transiciones | Moderada | Serio | Plantear la compra de una base de datos con mas capacidad y rendimiento |
| Proyecto y producto | Cambios repentinos en los requisitos | Moderada | Serio | Evaluar el impacto de los requisitos para ver si conviene o no |
| Negocio | Otra compañía lanza una aplicación con el mismo objetivo | Baja | Tolerable | Tratar de implementar más funciones llamativas sin que afecten al tiempo de entrega |
| Proyecto | Personal clave está enfermo o no esta disponible en momentos críticos | Moderada | Tolerable | Reorganizar el equipo para que haya más miembros que conzcan el trabajo del resto |
| Proyecto y producto | Componentes reutilizados limitan la funcionalidad | Muy baja | Insignificante | Modificar el código reutilizado para adaptarlo a las necesidades |

Herramientas usadas

Hasta la fecha

* Métodos de comunicación
  + Discord (Reuniones)
  + WhatsApp
* Trabajo colaborativo
  + Git/GitHub
  + Trello
* Elaboración de documentos
  + Microsoft Word
  + Pages
* Diseño gráfico
  + Adobe Photoshop
  + Affinity Designer
  + Pencil 2D
  + Procreate
* Diagrama de requisitos, clases y secuencia
  + MagicDraw
* Programación de la aplicación
  + Eclipse (Spring/Java)
  + Visual Studio Code (CSS y HTML)
* Diseño de presentación
  + Keynote
  + Jitter
  + iMovie

Requisitos

Fundamentales para el proyecto

Los requisitos son descripciones de los servicios que un sistema debe proporcionar y las restricciones a su modo de operación.

Un **requisito** es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Se usa en un sentido formal en diversas ingenierías.

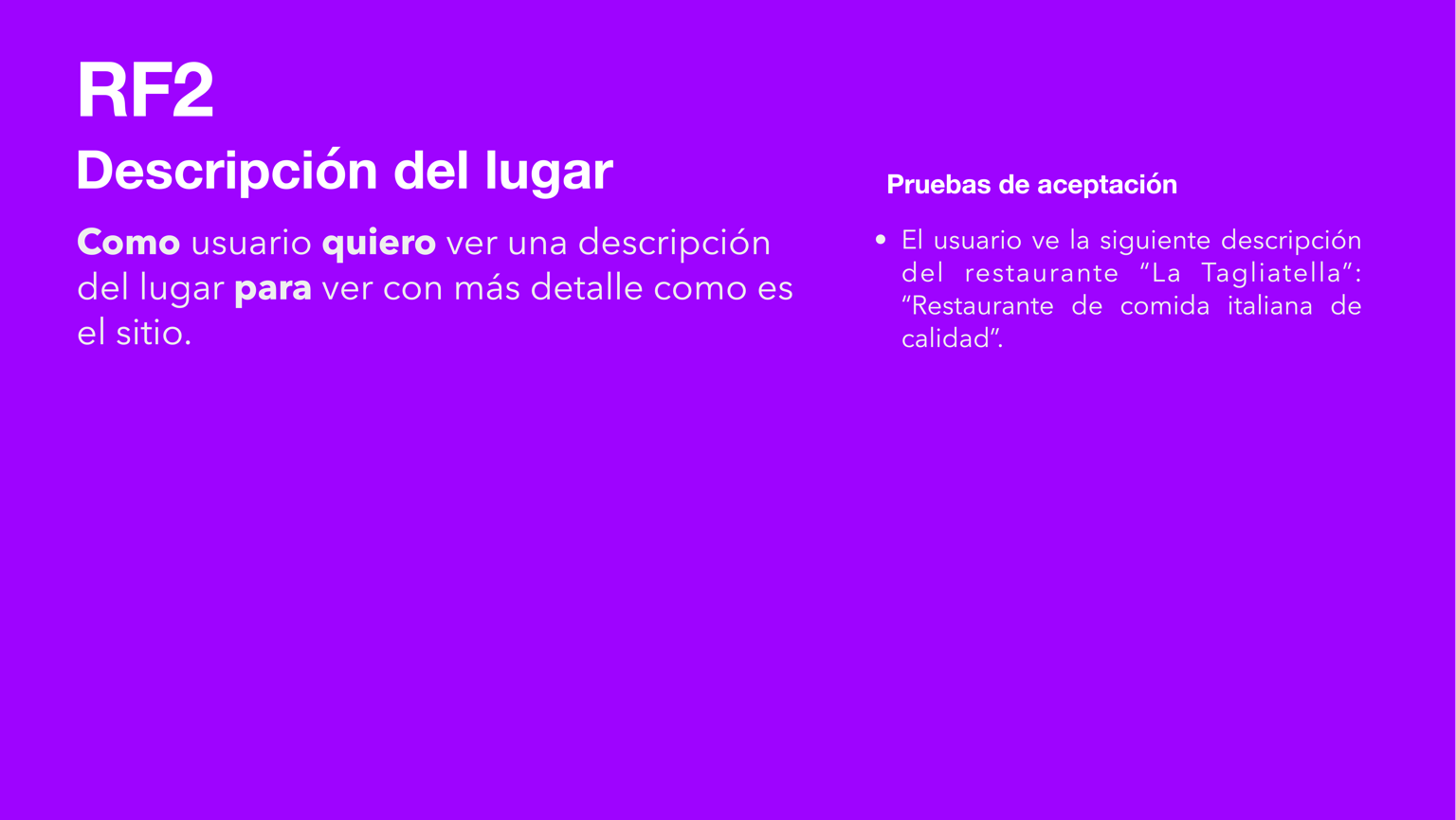
En este apartado vamos a mostrar los requisitos en forma de historias de usuario, ya que proporcionan un “enfoque ligero” para gestionar los requisitos de un sistema, son una pequeña pieza de funcionalidad que añade valor al negocio.

Los acompañamos con pruebas de aceptación, que detallan ejemplos de cómo interactuará cierto sujeto con la aplicación centrándose en el uso que le dará al requisito en cuestión.

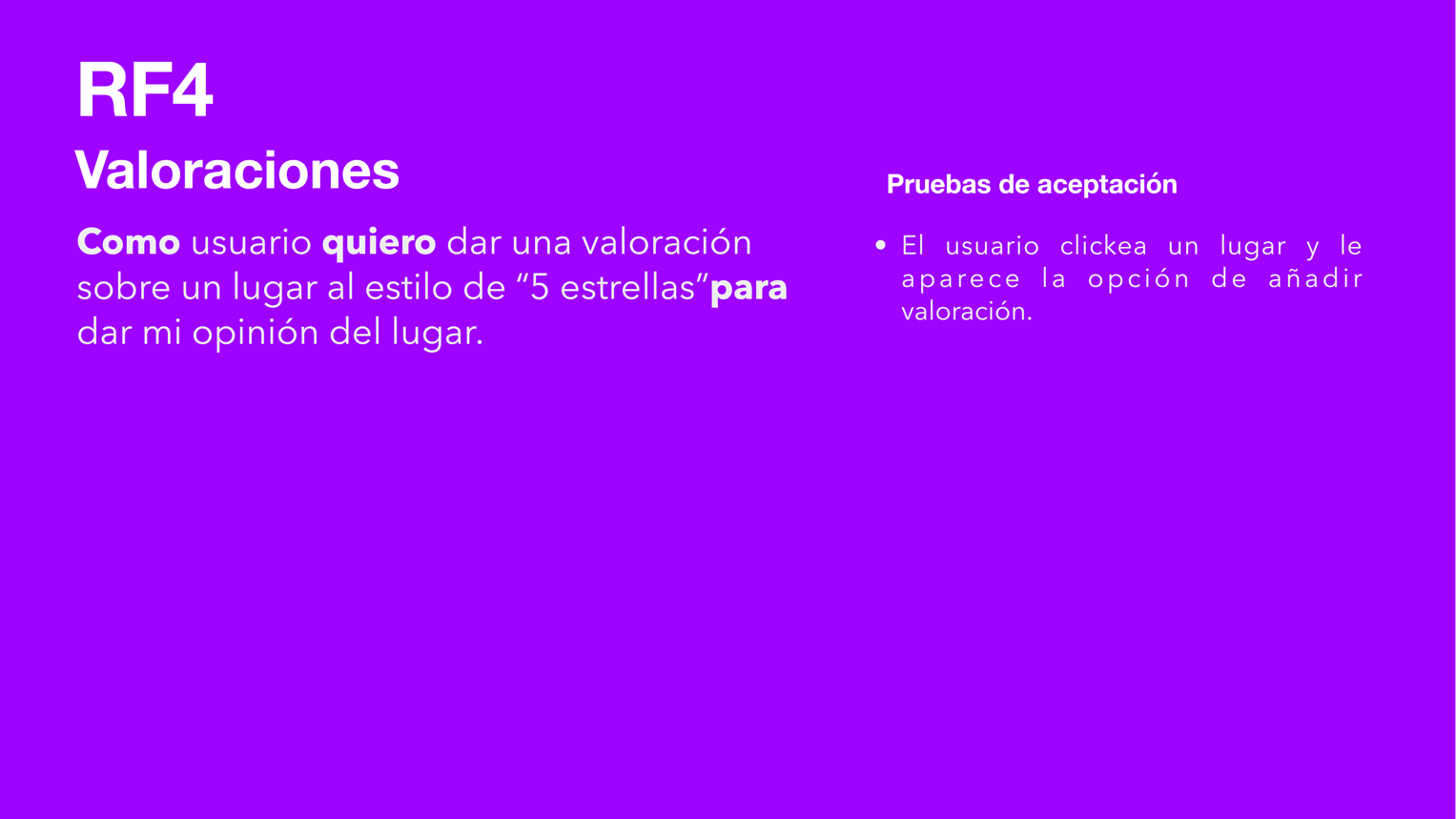
## Funcionales (RF)

Describen la funcionalidad o los servicios que se espera que el sistema suministre

.







Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza mediaTexto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

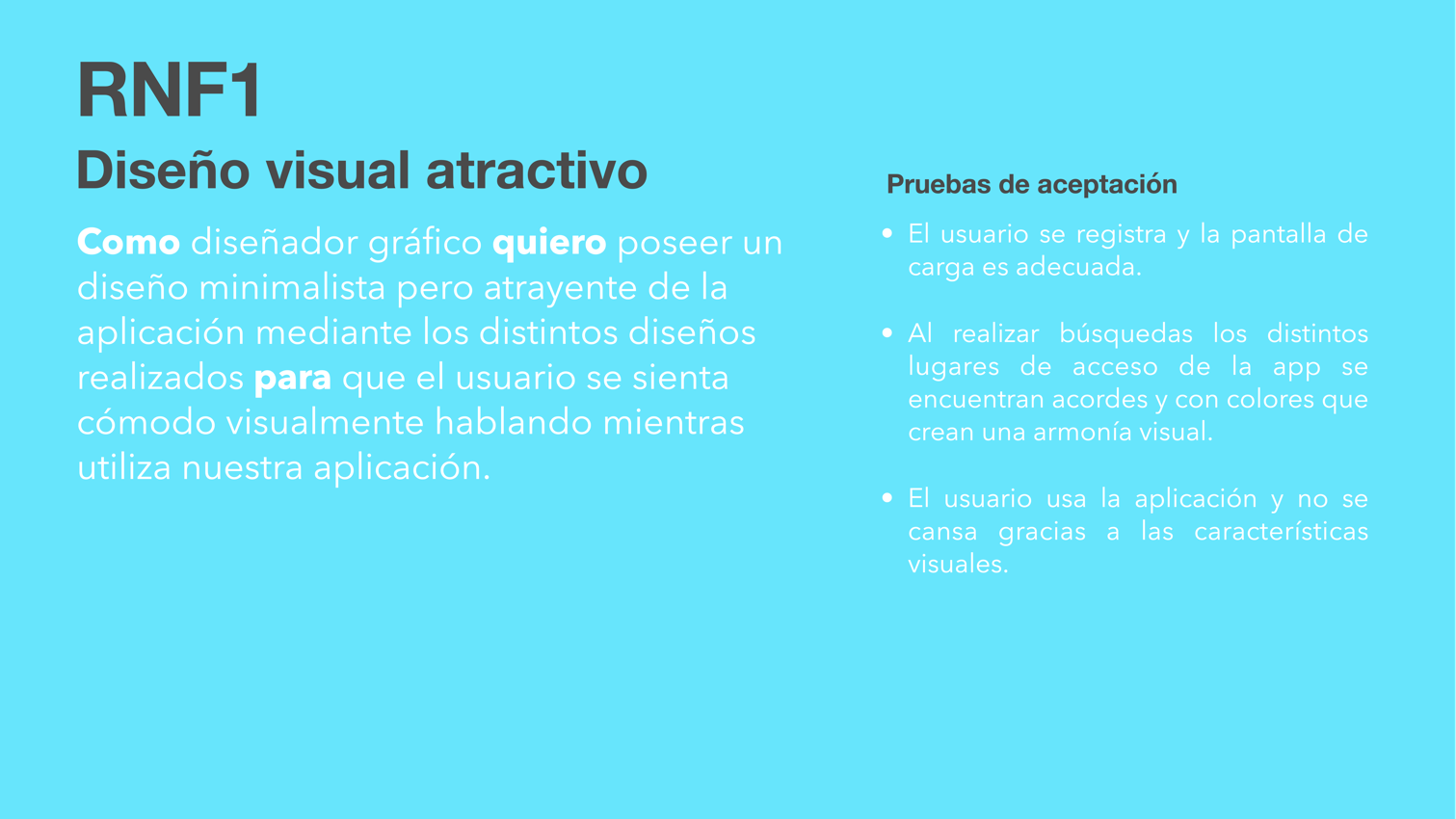
## Texto Descripción generada automáticamente

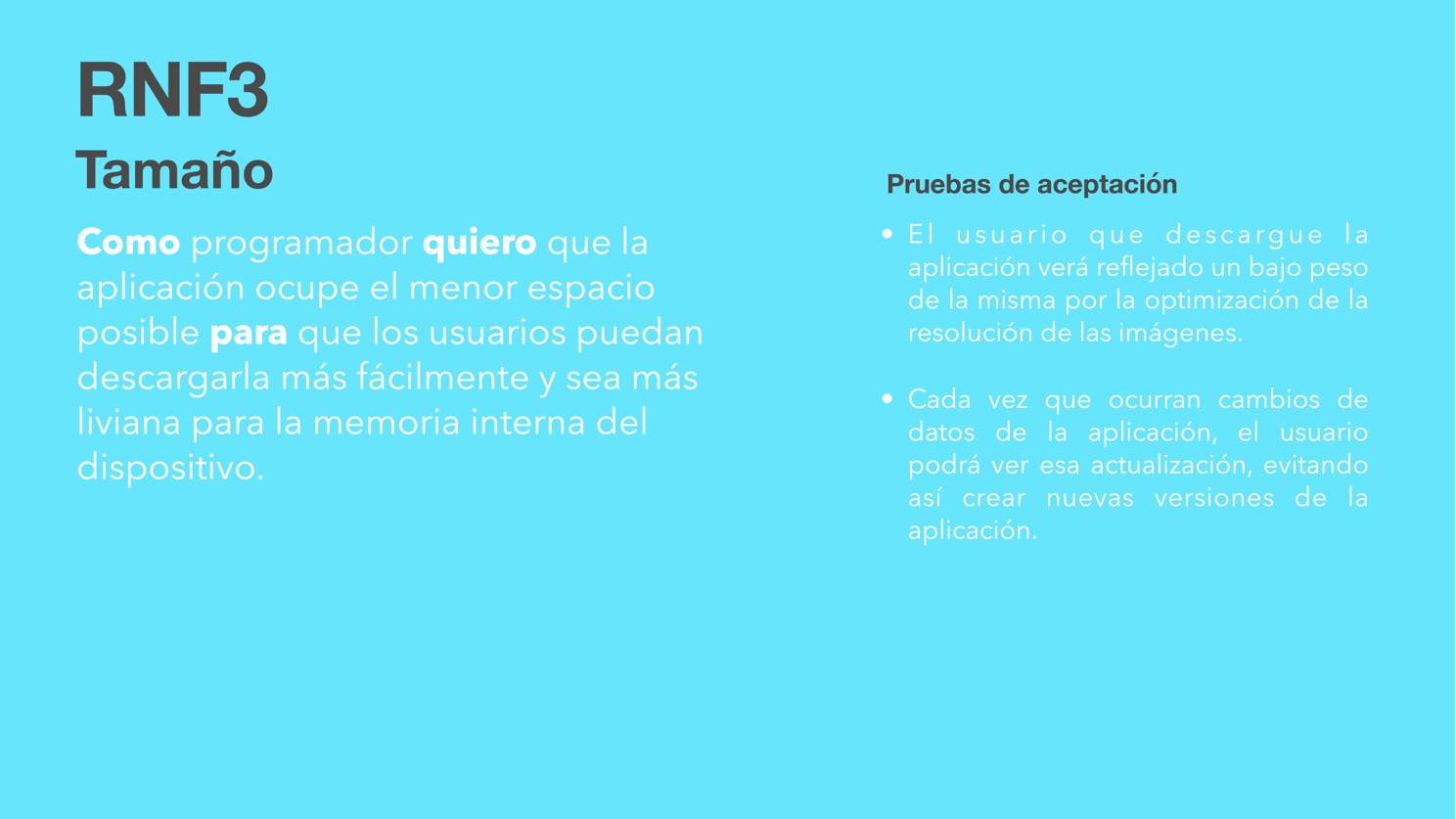
Texto

Descripción generada automáticamente

## Texto Descripción generada automáticamente

## No Funcionales (NRF)

****Restricciones sobre los servicios o funciones ofrecidas por el sistema.

****

## 

## Esquema de los requisitos

Diagrama

Descripción generada automáticamente

Esquema en MagicDraw de los requisitos y las relaciones entre ellos­­­

Casos de uso

Del software

Los diagramas de casos de uso muestran la funcionalidad del sistema desde el punto de vista de un observador externo.

## ACtores

* Usuario (el que usa la app)
* Administrador (Encargados de los aspectos técnicos)
* Moderadores (Conocedores de la ciudad, encargados de sugerir lugares y eliminar comentarios ofensivos)

## Casos de uso

* Usuario
  + Añadir una valoración o comentario
  + Añadir características del lugar, cuestionarios como google
  + Buscar nuevos planes para un día concreto
  + Sugerir nuevos lugares no puestos anteriormente (Han de ser aprobados)
* Administrador
  + Solucionar problemas puntuales
  + Aprobar y banear moderadores
* Moderador:
  + Añadir sugerencias semanales (Posibles campañas publicitarias).
  + Semanalmente, revisar lugares, comentarios y planes nuevos.
  + Actualizar sobre problemas o eventos en el lugar (Ej.: Obras de teatro, reparaciones en el lugar, etc.)

## cui1

**Identificador único:** Añadir una valoración o comentario.

**Contexto de uso:** Cuando un usuario quiere agregar su opinión sobre un sitio que ya haya visitado

**Precondiciones y activación:** El usuario está conectado y ha visitado el lugar recientemente.

**Garantías de éxito:** Se agrega una opinión con las huellas (estrellas) con o sin comentario de un lugar.

**Escenario principal:**

1. El usuario visita un lugar
2. El usuario quiere recomendar un sitio
3. El usuario busca el sitio
4. El usuario opina
5. Se pueblica

**Escenarios alternativos:** El moderador al cierto tiempo ve el comentario inadecuado y lo elimina.

## cui2

**Identificador único:** Añadir características del lugar, cuestionarios como google.

**Contexto de uso:** Un usuario recibe una notificación en el móvil para responder ciertas preguntas relacionadas con el sitio, tales como si tiene aparcamientos cerca, etc.

**Precondiciones y activación:** El usuario ha visitado el lugar recientemente y recibe una notificación para rellenar una encuesta

**Escenario principal:**

1. El usuario visita un lugar
2. El usuario recibe una notificación de encuesta
3. El usuario realiza la encuesta

**Escenarios alternativos:** El usuario decide no hacer la encuesta

## CUI3

**Identificador único:** Buscar nuevos planes para un día concreto.

**Contexto de uso:** El usuario quiere hacer planeas un día concreto y se mete en la app para buscar ideas.

**Precondiciones y activación:** El usuario desea buscar un lugar que visitar.

**Garantías de éxito:** La aplicación recomienda sitio al usuario.

**Escenario principal:**

1. El usuario entra en la APP.
2. Busca un tipo de sitio (restaurante, concierto o búsqueda general).
3. El usuario observa las distintas posibilidades.
4. El usuario va al sitio.

**Escenarios alternativos:** La app no va y no da resultados, el usuario no se decide por ningún plan que hay propuesto.

## Cui4

**Identificador único:** Sugerir nuevos lugares no puestos anteriormente (Han de ser aprobados).

**Contexto de uso:** El usuario agrega un lugar al que visitar no registrado en nuestra base de datos.

**Precondiciones y activación:** El usuario desea agregar ese lugar y le da a la pestaña de proponer sitio.

**Garantías de éxito:** Se recibe una petición de añadir ese sitio a la base de datos.

**Escenario principal:**

1. El usuario visita un sitio
2. Abre la app
3. Trata de aplicar para agregar un sitio
4. El moderador la recibe y la procesa
5. Se agrega el sitio

**Escenarios alternativos:** El lugar es un sitio falso, el lugar ya ha sido agregado.

## Cui5

**Identificador único:** Sugerir nuevos planes a un lugar (Han de ser aprobados).

**Contexto de uso:** El usuario agrega un plan que realizar no registrado en nuestra base de datos.

**Precondiciones y activación:** El usuario desea agregar ese plan y le da a la pestaña de proponer lugar.

**Garantías de éxito:** Se recibe una petición de añadir ese plana la base de datos.

**Escenario principal:**

1. El usuario visita un sitio.
2. Abre la app.
3. Trata de aplicar para agregar un sitio.
4. El moderador la recibe y la procesa.
5. Se agrega el sitio.

**Escenarios alternativos:** Sea un plan falso, ya este ese sitio agregado o sea un plan muy aburrido y el moderador no lo acepte.

## Cui6

**Identificador único:** Solucionar problemas puntuales.

**Contexto de uso:** Se notifica un error en la aplicación web y el administrador se dispone a solucionarlo.

**Precondiciones y activación:** Hay un error notable.

**Garantías de éxito:** Ese error no se repite más.

**Escenario principal:**

1. Se recibe una advertencia de error.
2. El moderador la procesa.
3. Se soluciona el error.

**Escenarios alternativos:** los administrador no son capaces de resolver ese error.

## Cui7

**Identificador único:** Aprobar y banear moderadores.

**Contexto de uso:** Se necesitan moderadores para tener la app en buen estado.

Garantías de éxito: Se buscan moderadores suficientes.

**Escenario principal:**

1. Nos damos cuenta de que hay muchas peticiones y propuestas.
2. Buscamos moderadores.
3. Se lee las diferentes aplicaciones desde nuestra app.
4. Se aceptan ciertas solicitudes.

**Escenarios alternativos:** No encontremos moderadores aptos para la tarea.

## Cui8

**Identificador único:** Añadir sugerencias semanales (Posibles campañas publicitarias).

**Contexto de uso:** Promocionar nuevas actividades para la app.

**Garantías de éxito:** La app tiene más movimiento..

**Escenario principal:**

1. El moderador abre la app.
2. Crea ciertos eventos (en base a valoraciones anteriores o a raíz de campañas publicitarias).
3. la gente consulta esas sugerencias y las toma.

**Escenarios alternativos:** Las sugerencias son deficientes o al moderador se le olvida realizar estas sugerencias.

## Cui9

**Identificador único:** Semanalmente, revisar lugares, comentarios y planes nuevos.

**Contexto de uso:** Revisar las diferentes peticiones de los usuarios.

**Garantías de éxito:** La bandeja de peticiones este vacía.

**Escenario principal:**

1. Cada ciertos días el moderador abre su bandeja de propuestas.
2. Las analiza y ve cuales son factibles y cuales no.
3. Las agrega a la app.
4. Acepta las opiniones.
5. Descarta las opiniones o lugares o planes malos por diferentes motivos (ya mencionados antes).

**Escenarios alternativos:** Sean tantas las sugerencias que sea imposible analizarlas todas.

## Cui10

**Identificador único:** Actualizar sobre problemas o eventos en el lugar (Ejemplos: Obras de teatro, reparaciones en el lugar, etc.)

**Contexto de uso:** Había un evento externo el cual impedía acceder a cierto sitio.

**Garantías de éxito:** Se actualizó.

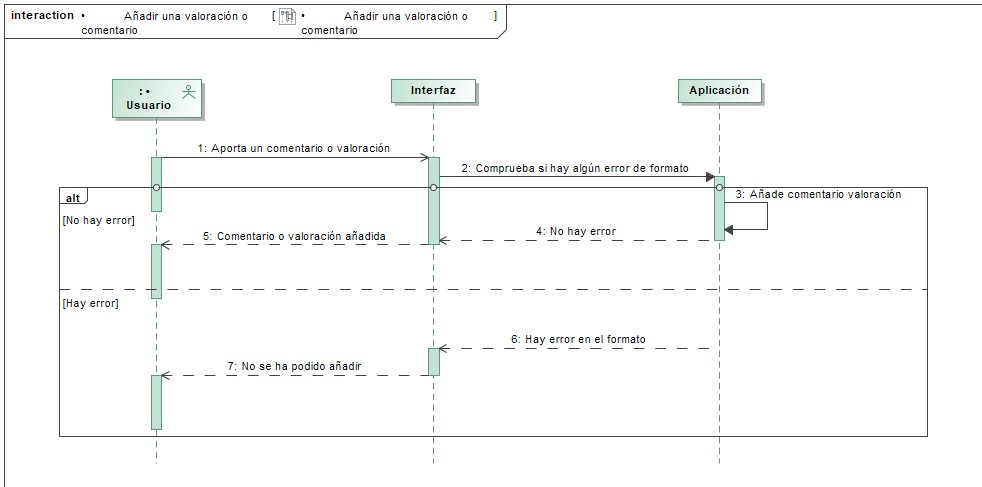
**Escenario principal:**

1. previamente se había agregado un problema que impedía realizar cierto plan.
2. Se acaba este problema.
3. El moderador entra y borra el evento el cual molestaba.
4. Se puede volver a realizar ese plan.

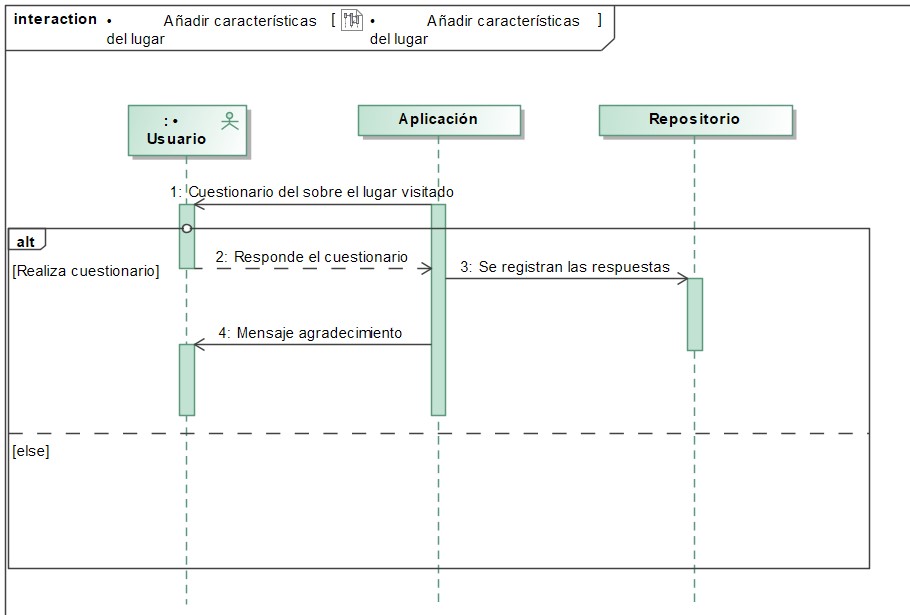
**Escenarios alternativos:** Se le olvide borrar al moderador.

## Diagrama de casos de uso

## Diagramas de secuencia

Los diagramas de secuencia muestran la interacción de un conjunto de objetos enfatizando el orden en el tiempo. Permite refinar los casos de uso incluyendo detalles de implementación como los objetos y clases usados y los mensajes intercambiados entre los mismos.

(CUI 1) El usuario accede al lugar visitado recientemente y en el apartado de valoraciones y comentarios dejara sus respectivas opiniones sobre el lugar, siendo la valoración permitida de 1 a 5. Si hay algún error de formato como que el usuario intente añadir un comentario vacío, este se descartará

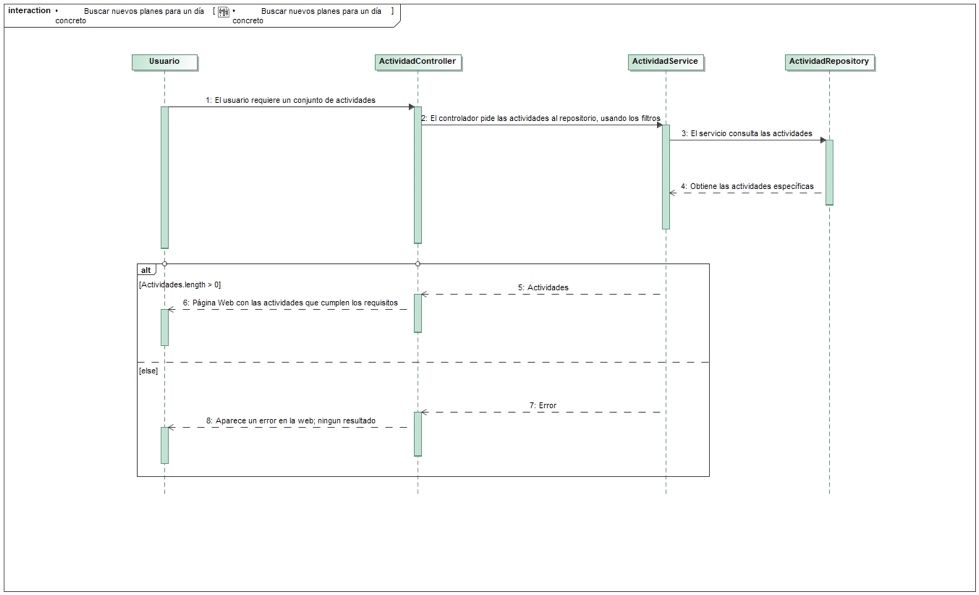
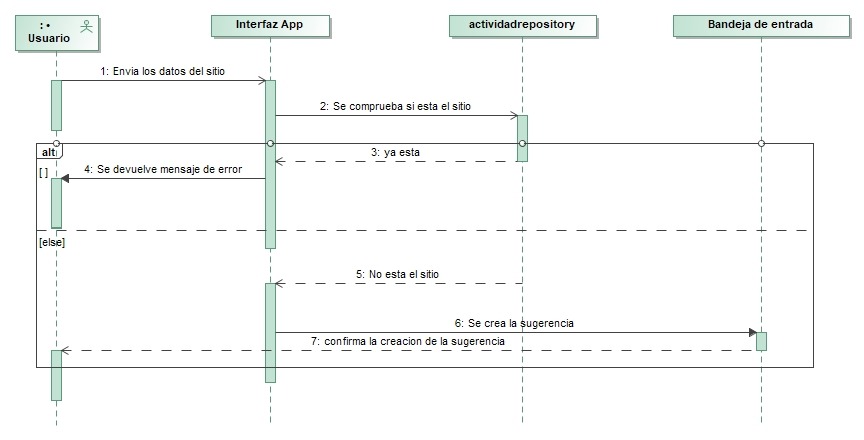


(CUI 2) El usuario recibe una notificación del lugar que ha visitado gracias a las recomendaciones de la aplicación,

se le facilita un cuestionario en el que hay preguntas sobre el lugar visitado.

Si el usuario responde correctamente, se registran las respuestas y se agradece la colaboración al usuario,

por el contrario, si el usuario ignora la notificación, no se hará nada al respecto.

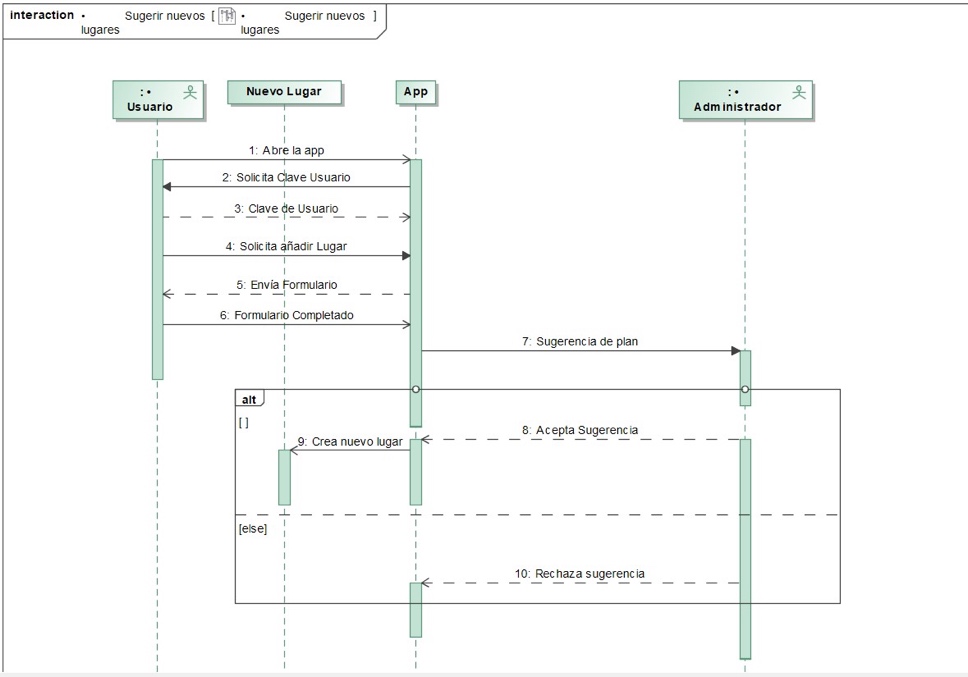


(CUI 4) 1. El usuario enviará los datos por medio de la interfaz de la app, desde aquí se consulta en la base si el sitio ya está agregado o no, el cual más tarde se envía a la bandeja de entrada de solicitudes que más tarde se administrará

2. Si ya está agregado se le dice al usuario que ese sitio ya está.

(CUI 3) El usuario pedirá al ActividadController que obtenga las actividades, solicitando a ActividadService que el repositorio de actividades (ActividadRepository) le devuelva un conjunto de actividades que cumplan el filtro elegido por el usuario, en este caso cumpliendo el requerimiento de que sea apto para ese día.

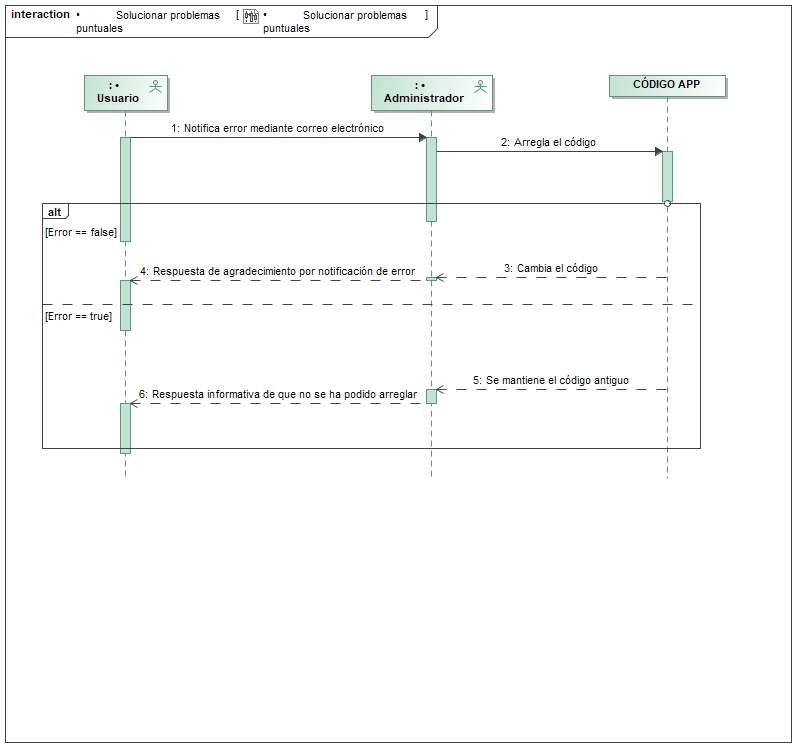
Si el conjunto devuelto es vacío, se mostrará un error por pantalla pero si por el contrario hay elementos, se mostraran en ella.



(CUI 5) El usuario inicia sesión en la app para añadir un nuevo lugar en el que ha estado,

la app le proporciona un formulario y cuando este lo rellena el administrador decide

si aceptar o no la sugerencia. Si es aceptada, la app crea el nuevo lugar.



(CUI 6) El usuario notifica un error que haya encontrado en la aplicación mediante correo electrónico.

El correo llega al administrador, que intenta arreglar el código de la app. En caso exitoso,

tras probar que funciona, se le responde al cliente con un correo agradeciendo su notificación.

En caso contrario, le informamos de que no hemos sido capaces de arreglar el error.

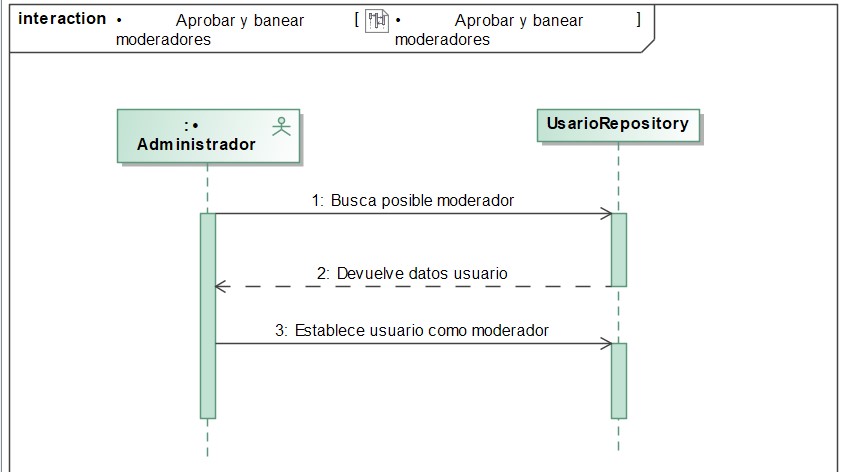
(CUI 7) El administrador pide un usuario, el cual le entregará el UsuarioController, que para encontrarlo

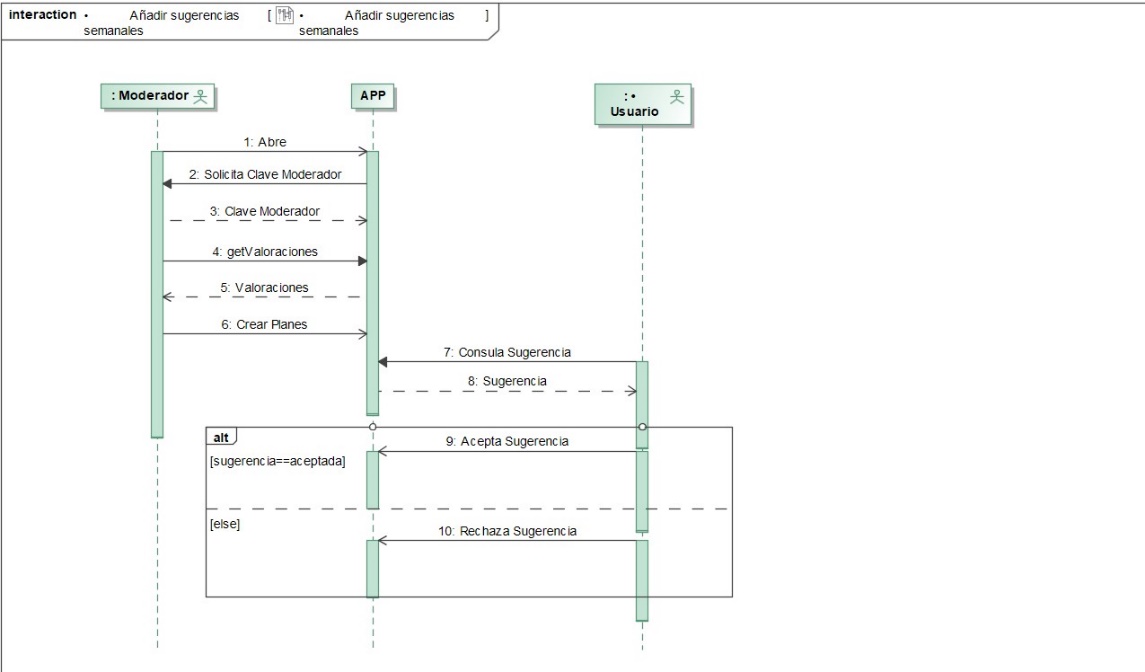
tiene que pedirle al UsuarioService que acceda al repositorio de usuarios y encuentre al susodicho.

Una vez encontrado (o no) entrega al UsuarioController el usuario. Si le manda al administrador el

usuario pedido, éste procederá a crear o eliminar (según el contexto de la situación) una cuenta de

moderador. En caso de que no le llegue ningún usuario, el administrador no podrá hacer nada.

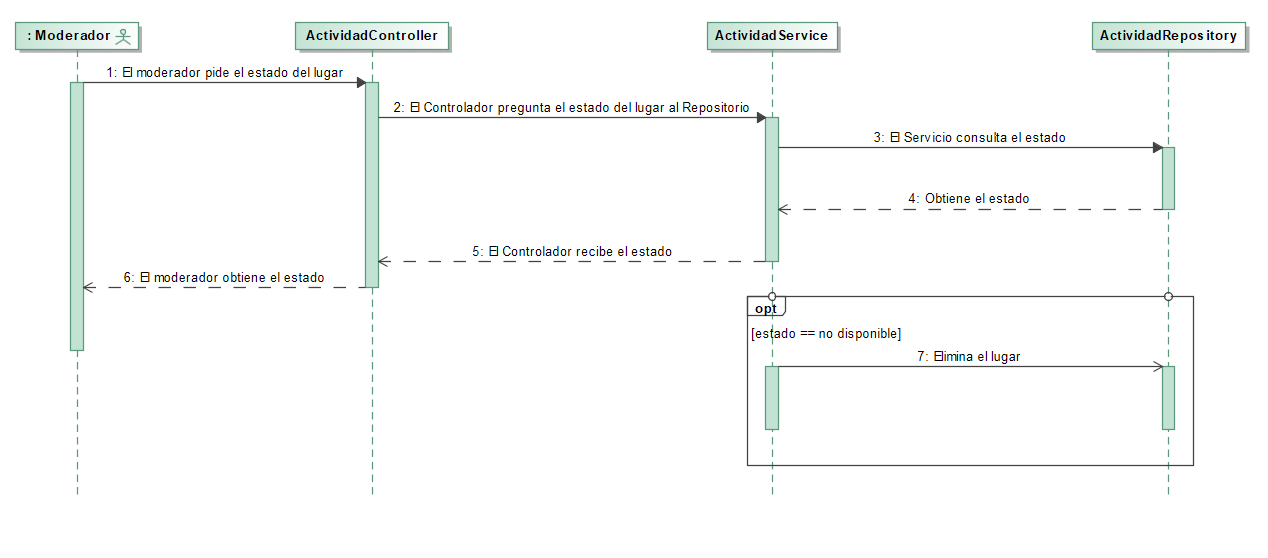
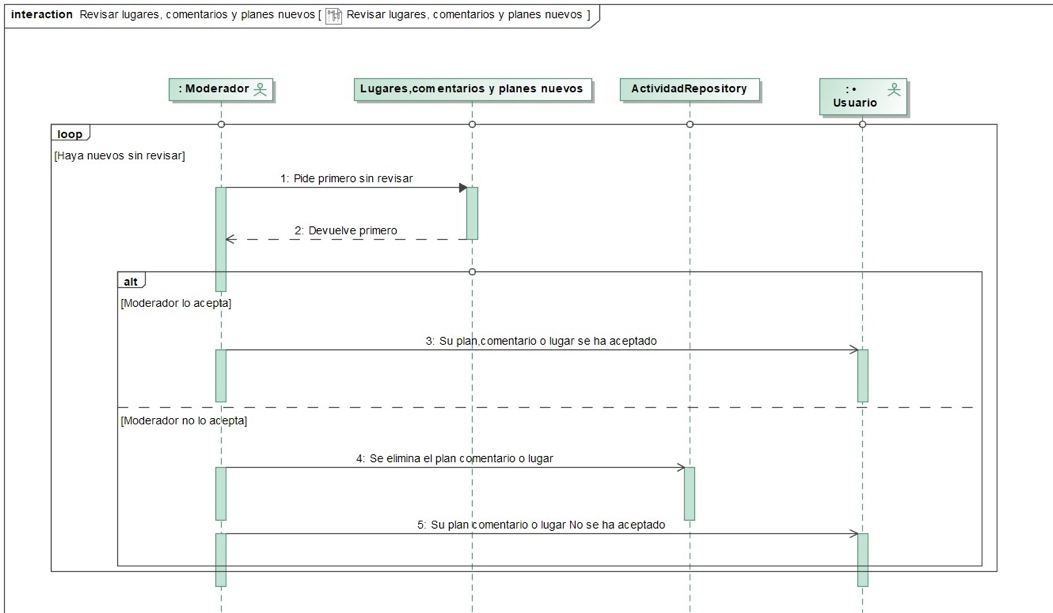




(CUI 8) 1. El moderador entra a la aplicación y en función de las sugerencias recibidas creará planes.

2. El cliente consulta las sugerencias creadas.

3. El cliente decidirá si aceptar o no las sugerencias recomendadas.



(CUI 10) 1. El moderador pide el estado del lugar al ActividadController

2. El Controlador pregunta por el estado al Repositorio

3. El ActividadService consulta en el Repositorio el estado

4. Se obtiene el estado

5. El Controlador recibe el estado y se lo pasa al Moderador

6. El Moderador recibe el estado y de esta manera sabe lo que tiene que hacer

7. Si el estado es "no disponible" se borra el lugar del repositorio, sino no se hace nada

(CUI 9) 1. Es un bucle, el moderador solicita el primer mensaje de la bandeja de entrada, esta se la devuelve, el administrador comprueba si es válida, la implementa en los datos, devuelve que todo está correcto.

2. Si no es correcta no se agrega y se le envía un mensaje al usuario sobre el motivo.

3. Si no hay suficiente espacio se le dice al moderador

Diagrama de clases

Diagrama

Descripción generada automáticamente

Diagrama de estructura estática que muestra las clases del sistema y sus interrelaciones

Pruebas junit

Para comprobar el correcto funcionamiento de una unidad del código de nuestras clases principales (actividad, comentario y usuario) hemos realizado las siguientes pruebas Junit.

Además para algunas de ellas hemos utilizado Mockitos, para simular la creación de algunos objetos.

Texto

Descripción generada automáticamente