Ingeniería del Software

Grado de Ingeniería Informática
Curso 2020-2021
Antonio Moruno Gracia, David Perez Dueñas, Marcos Rivera Gavilán

Práctica 2

Proyecto 26

1. Introducción

La práctica 2 se ha realizado a lo largo de 3 semanas concretamente del 8 al 22 de octubre de 2020. Para esta práctica hemos tenido que realizar el análisis de requisitos de un sistema de gestión de la información de parques naturales, para una empresa de gestión de estos entornos en Andalucía. Esta empresa es la encargada de registrar los espacios que gozan de este reconocimiento, garantizando su mantenimiento y conservación, así como programar rutas por sus senderos.

Los objetivos a cumplir con la realización de la práctica eran los siguientes:

- Aprender a extraer requisitos a partir de la entrevista con el cliente.
- Identificar, refinar y documentar los distintos tipos de requisitos del sistema.
- Utilizar historias de usuario para crear la lista de producto según la metodología Scrum.
- Comenzar a planificar y priorizar funcionalidades de cara a su futura implementación
- Aplicar la técnica de casos de uso de UML para detallar el análisis de requisitos funcionales. Uso de Visual Paradigm.
- Validar los requisitos con el cliente para resolver dudas surgidas durante la toma de requisitos.

2. Planificación

En esta práctica le ha tocado ser Product Owner a Marcos Rivera Gavilán y lo largo de estas tres semanas, hemos distribuido el trabajo de la siguiente forma:

2.1 Semana 1

El trabajo que debíamos realizar para esta semana era el siguiente:

- Extraer requisitos a partir de la entrevista con el cliente.
- Identificar, refinar y documentar los distintos tipos de requisitos del sistema.

Y acordamos distribuirlo de la siguiente forma:

- Antonio: Requisitos funcionales
- Marcos: Requisitos de información
- David: Requisitos no funcionales

Siendo cada uno responsable de:

- Añadir y documentar sus requisitos en el documento de Requisitos del Sistema

2.2 Semana 2

El trabajo que debíamos realizar para esta semana era el siguiente:

- Utilizar historias de usuario para crear la lista de producto según la metodología Scrum.
- Comenzar a planificar y priorizar funcionalidades de cara a su futura implementación.

Y acordamos distribuirlo de la siguiente forma:

- Antonio: Análisis de los requisitos funcionales para obtener historias de usuario
- Marcos: Análisis de los requisitos de información para obtener historias de usuario
- David: Análisis de los requisitos no funcionales para obtener historias de usuario

Siendo cada uno responsable de:

- Añadirlas al tablón haciendo uso de las funcionalidad de historias de usuario
- Añadirlas al documento Historias de Usuario

2.3 Semana 3

El trabajo que debíamos realizar para esta semana era el siguiente:

- Aplicar la técnica de casos de uso de UML para detallar el análisis de requisitos funcionales. Uso de Visual Paradigm.
- Validar los requisitos con el cliente para resolver dudas surgidas durante la toma de requisitos

Y acordamos distribuirlo de la siguiente forma:

- Antonio: Creación de los casos de uso
- Marcos: Validación de los requisitos y redacción de este documento.
- David: Diseño del diagrama UML

Siendo cada uno responsable de aportar su parte en tiempo y forma antes de la fecha de límite para la entrega de la práctica

El enlace a la instancia YouTrack es el siguiente:

https://uco-is2021-eq26.myjetbrains.com/

3. Especificación de requisitos

En esta sección procederemos a describir los requisitos del sistema que se extrajeron el pasado jueves 8 de octubre en la reunión.

3.1. Requisitos funcionales

RF1. Automatización de la gestión

El sistema debe automatizar toda la gestión de la información de los parques naturales de Andalucía abiertos a los visitantes.

RF2. Información

El sistema debe suministrar a los administrativos la información sobre la organización de las rutas y el estado de las diferentes rutas

RF3. Conexión con el sistema de nóminas

El sistema debe ser compatible con el sistema de nóminas de los monitores para lo cual debemos mantener un cómputo de las horas que trabaja cada monitor.

RF4. Reserva de rutas

El sistema debe permitir al administrador la reserva y la cancelación de una ruta que un usuario solicite hacer, siempre y cuando queden más de 15 minutos para el inicio de la misma.

RF5. Diferenciación de funcionalidades

Los monitores solo pueden tener acceso a las funcionalidades del software relativas a la creación y administración de las rutas, sin embargo los administrativos deben poder gestionar todo el sistema.

3.2. Requisitos de información

RI1. Información sobre los parques

El sistema debe almacenar la siguiente información sobre los parques naturales: identificador, nombre, localización en el mapa, municipio, provincia, superficie, fecha en la que se le declaró parque natural y sus premios, horario.

RI2. Información sobre los senderos

El sistema debe almacenar la siguiente información sobre los senderos: identificador, nombre, nivel de dificultad, estado del sendero y parque al que pertenece.

RI3. Información sobre las rutas

El sistema debe almacenar la siguiente información sobre las rutas: identificador, nombre, monitor que las dirige y suplente, modalidad a pie o en bici, longitud, duración de la ruta, pueden ser exclusivas ej. solo colegios y número de plazas, participantes inscritos y sus necesidades.

RI4. Información sobre los monitores

El sistema debe almacenar la siguiente información sobre los monitores: nombre y apellidos, DNI, fecha de nacimiento, teléfono de contacto, dirección, correo electrónico, horas trabajadas.

RI5. Información sobre los administrativos

El sistema debe almacenar la siguiente información sobre los administrativos: nombre y apellidos, DNI, fecha de nacimiento, dirección, correo electrónico.

RI6. Información sobre los senderistas

El sistema debe almacenar la siguiente información sobre los senderistas que participan en las rutas: nombre y apellidos, fecha de nacimiento, DNI, teléfono de contacto y si tienen requisitos especiales ej. movilidad reducida, ruta en la que participan.

3.3. Requisitos no funcionales

RNF1. Lenguaje

El programa debe de estar escrito en C++.

RNF2. Interfaz

La interfaz del programa será la terminal.

RNF3. Usabilidad

El sistema debe de estar operativo para todas los parques en horario de oficina (9:00-18:00). El sistema no debe de exceder los 5 segundos de caída.

RNF4. Acceso

Los empleados del parque deben de identificarse para acceder al sistema

RNF5. Normativa

Los datos de los senderistas deben de ser tratados conforme a la GPRD

4. Análisis de requisitos

4.1 Historias de usuario

En esta subsección procederemos a describir las historias de usuario en base a los requisitos que se extrajeron el pasado jueves 8 de octubre en la reunión.

HU1. COMO administrativo QUIERO administrar los parques PARA actualizar la información del sistema

Los administrativos deben poder modificar la información almacenada sobre los parques.

Haciendo uso del identificador del parque los administrativos podrán añadir, modificar, eliminar y consultar información sobre los mismos.

HU2. COMO administrativo QUIERO administrar los senderos PARA actualizar la información del sistema

Los administrativos deben poder modificar la información almacenada sobre los senderos.

Haciendo uso del identificador del parque en el que se encuentra dicho sendero y de su identificador los administrativos podrán añadir, modificar, eliminar y consultar información sobre los mismos.

HU3. COMO administrativo/monitor QUIERO administrar las rutas PARA actualizar la información del sistema

Los administrativos y los monitores deben poder modificar la información almacenada sobre las rutas.

Haciendo uso del identificador del parque en el que se encuentra dicha ruta y del identificador de la ruta los trabajadores podrán añadir, modificar, eliminar y consultar información sobre la misma.

HU4. COMO administrativo QUIERO administrar la información de los senderistas PARA actualizar la información del sistema

Los administrativos deben poder modificar la información almacenada sobre los senderistas.

Haciendo uso del dni del senderista los administrativos podrán añadir, modificar, eliminar y consultar información sobre los mismos.

HU5. COMO administrativo QUIERO administrar la plantilla PARA actualizar la información del sistema

Los administrativos deben poder modificar la información almacenada sobre la plantilla.

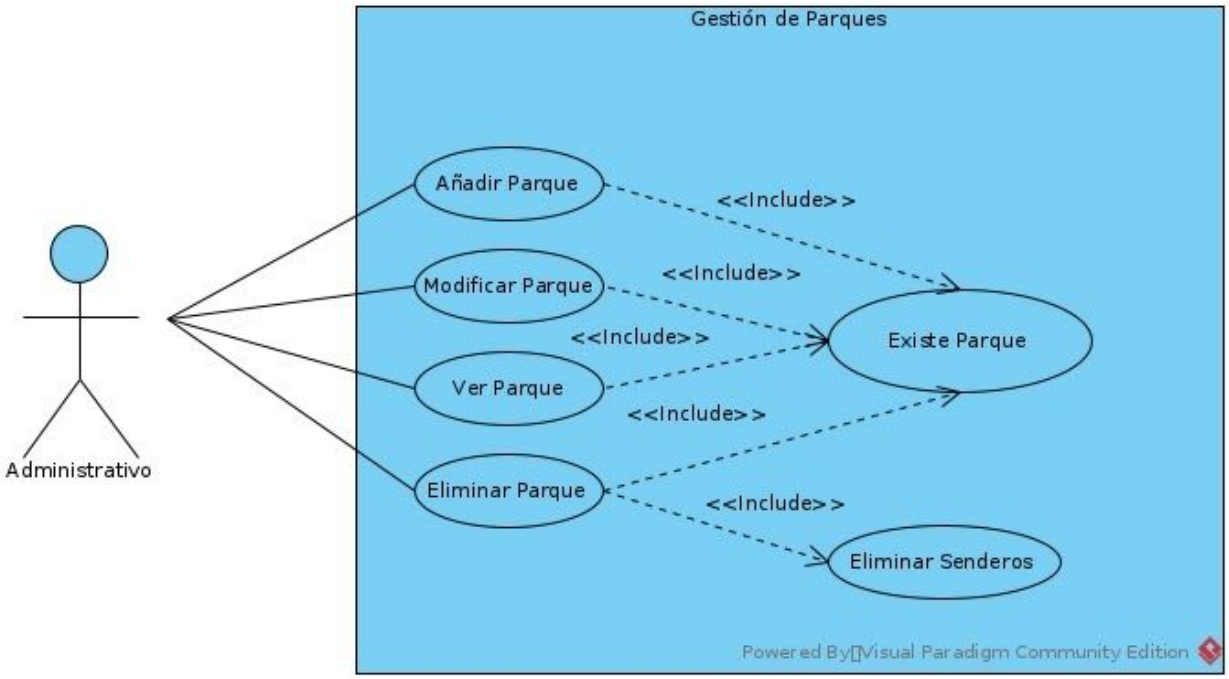
Haciendo uso del dni del trabajador los administrativos podrán añadir, modificar, eliminar y consultar información sobre los empleados.

HU6. COMO administrativo QUIERO poder obtener las horas de trabajo de los monitores PARA pasar la información al sistema de nóminas

Los administrativos deben poder acceder al cómputo de horas de los monitores para poder trasladar esta información al sistema de gestión de nóminas.(En un futuro es posible que esta tarea se automatice así que la información de salida ha de ser compatible con el sistema actual)

4.2 Casos de uso

4.2.1 Gestión de parques



4.2.1.1 CU1

Caso de uso: El administrativo quiere añadir un parque al sistema

Objetivo: Administrar los parques de forma que se pueda añadir información al sistema

Contexto: El parque no debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir los datos del parque
2. El sistema debe almacenar la nueva información

Extensiones:

1. a El sistema debe comprobar que el parque no exista(CU2)

4.2.1.2 CU2

Caso de uso: Comprobar si un parque existe en el sistema

Objetivo: Evitar la repetición de la información en el sistema

Contexto: Debe de existir como mínimo un parque en el sistema

Actor principal: Sistema

Escenario principal:

1. El administrativo debe introducir los datos del parque cuya existencia queremos comprobar
2. El sistema debe comparar la información introducida con la información disponible
3. El sistema debe devolver TRUE si el parque se encuentra en el sistema y FALSE si no se encuentra

4.2.1.3 CU3

Caso de uso: El administrativo quiere modificar un parque del sistema

Objetivo: Administrar los parques de forma que se pueda actualizar la información del sistema

Contexto: El parque debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir los datos del parque
2. El administrativo debe modificar los datos del parque que desee
3. El sistema debe almacenar la nueva información

Extensiones:

1. a El sistema debe comprobar que el parque exista(CU2)

4.2.1.4 CU4

Caso de uso: El administrativo quiere eliminar un parque del sistema

Objetivo: Administrar los parques de forma que se pueda eliminar información del sistema

Contexto: El parque debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir los datos del parque
2. El sistema debe eliminar la información del parque

Extensiones:

1. a El sistema debe comprobar que el parque exista(CU2)
2. a El sistema debe borrar la información de los senderos asociados a dicho parque(CU9)

4.2.1.5 CU5

Caso de uso: El administrativo quiere ver la información de un parque del sistema

Objetivo: Administrar los parques de forma que se pueda visualizar la información del sistema

Contexto: El parque debe de existir en el sistema

Actor principal: Administrativo

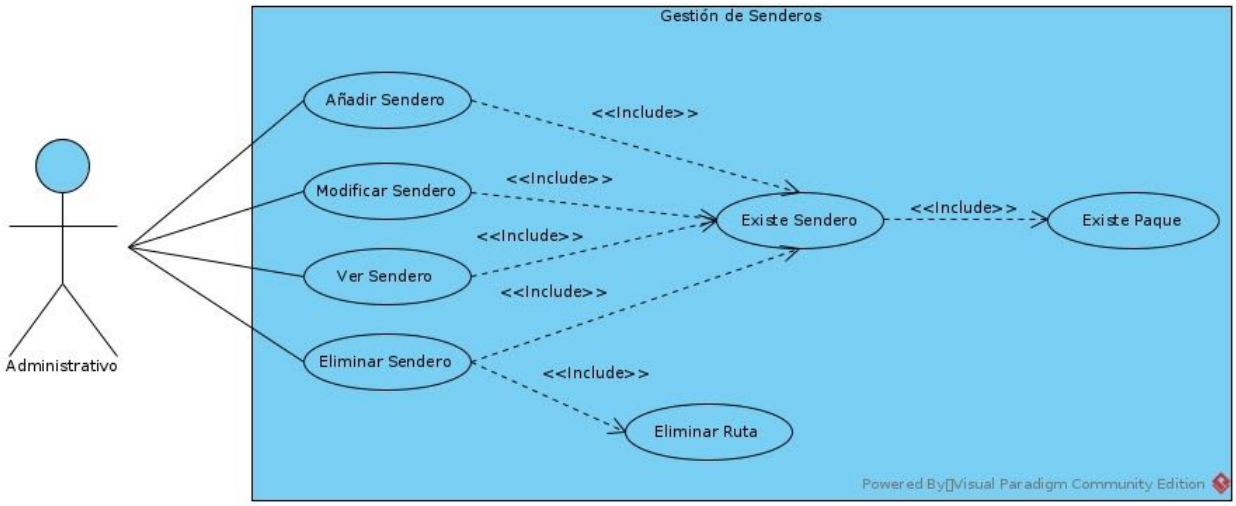
Escenario principal:

1. El administrativo debe introducir los datos del parque
2. El sistema debe mostrar la información

Extensiones:

1. a El sistema debe comprobar que el parque exista(CU2)

4.2.2 Gestión de senderos



4.2.2.1 CU6

Caso de uso: El administrativo quiere añadir un sendero a un parque

Objetivo: Administrar los senderos de forma que se puedan añadir senderos a los parques del sistema

Contexto: El sendero no debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir los datos del sendero y del parque al que pertenece
2. El sistema debe almacenar la nueva información

Extensiones:

1. a El sistema debe comprobar que el sendero no exista (CU7)

4.2.2.2 CU7

Caso de uso: Comprobar si un sendero pertenece a un parque

Objetivo: Evitar la repetición de la información en el sistema

Contexto: Debe de existir como mínimo un parque y un sendero en el sistema

Actor principal: Sistema

Escenario principal:

1. El administrativo debe introducir los datos del sendero cuya existencia queremos comprobar
2. El sistema debe comparar la información introducida con la información disponible
3. El sistema debe devolver TRUE si el sendero se encuentra en dicho parque y FALSE si no se encuentra

4.2.2.3 CU8

Caso de uso: El administrativo quiere modificar un sendero de un parque

Objetivo: Administrar los senderos de forma que se pueda actualizar la información del sistema

Contexto: El sendero debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir los datos del sendero
2. El administrativo debe modificar los datos del sendero que desee
3. El sistema debe almacenar la nueva información

Extensiones:

1. a El sistema debe comprobar que el sendero exista(CU7)

4.2.2.4 CU9

Caso de uso: Eliminar un sendero del sistema

Objetivo: Administrar los senderos de forma que se pueda eliminar información del sistema

Contexto: El sendero debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir los datos del sendero
2. El sistema debe eliminar la información del sendero

Extensiones:

1. a El sistema debe comprobar que el sendero exista(CU7)
2. a El sistema debe borrar la información de las rutas asociados a dicho sendero(CU14)

4.2.2.5 CU10

Caso de uso: El administrativo quiere ver la información de un sendero del sistema

Objetivo: Administrar los senderos de forma que se pueda visualizar la información del sistema

Contexto: El sendero debe de existir en el sistema

Actor principal: Administrativo

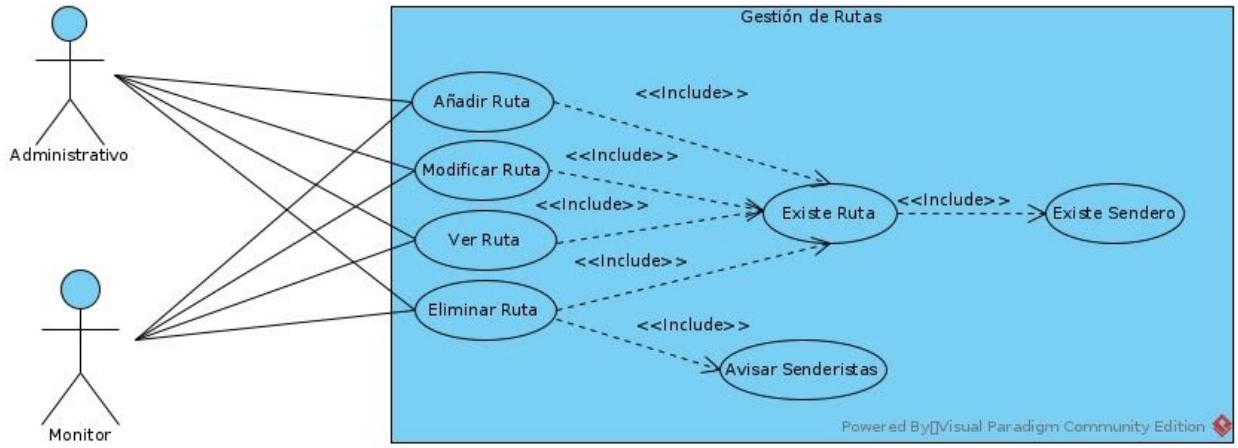
Escenario principal:

1. El administrativo debe introducir los datos del sendero
2. El sistema debe mostrar la información

Extensiones:

1. a El sistema debe comprobar que el sendero exista(CU7)

4.2.3 Gestión de rutas



4.2.3.1 CU11

Caso de uso: El administrativo o el monitor quiere añadir una ruta

Objetivo: Administrar las rutas de forma que se puedan añadir rutas al sistema

Contexto: La ruta no debe de existir en el sistema

Actor principal: Administrativo o monitor

Escenario principal:

1. El administrativo o el monitor debe introducir los datos de la ruta
2. El sistema debe almacenar la nueva información

Extensiones:

1. a El sistema debe comprobar que la ruta no exista (CU12)

4.2.3.2 CU12

Caso de uso: Comprobar si una ruta existe

Objetivo: Evitar la repetición de la información en el sistema

Contexto: Debe de existir como mínimo un parque, un sendero y una ruta en el sistema

Actor principal: Sistema

Escenario principal:

1. El administrativo o el monitor debe introducir los datos de la ruta cuya existencia queremos comprobar
2. El sistema debe comparar la información introducida con la información disponible
3. El sistema debe devolver TRUE si la ruta existe y FALSE si no existe

4.2.3.3 CU13

Caso de uso: El administrativo o el monitor quiere modificar una ruta

Objetivo: Administrar las rutas de forma que se pueda actualizar la información del sistema

Contexto: La ruta debe de existir en el sistema

Actor principal: Administrativo o monitor

Escenario principal:

1. El administrativo o monitor debe introducir los datos de la ruta
2. El administrativo o monitor debe modificar los datos de la ruta que desee
3. El sistema debe almacenar la nueva información

Extensiones:

1. a El sistema debe comprobar que la ruta exista(CU12)

4.2.3.4 CU14

Caso de uso: Eliminar una ruta del sistema

Objetivo: Administrar las rutas de forma que se pueda eliminar información del sistema

Contexto: La ruta debe de existir en el sistema

Actor principal: Administrativo o monitor

Escenario principal:

1. El administrativo o monitor debe introducir los datos de la ruta
2. El sistema debe eliminar la información de la ruta

Extensiones:

1. a El sistema debe comprobar que la ruta exista(CU12)
2. a El sistema debe recordarle al administrador o monitor avisar a los excursionistas apuntados a dicha ruta(CU28)

4.2.3.5 CU15

Caso de uso: El administrativo o monitor quiere ver la información de una ruta

Objetivo: Administrar las rutas de forma que se pueda visualizar la información del sistema

Contexto: La ruta debe de existir en el sistema

Actor principal: Administrativo o monitor

Escenario principal:

1. El administrativo o monitor debe introducir los datos de la ruta
2. El sistema debe mostrar la información

Extensiones:

1. a El sistema debe comprobar que la ruta exista(CU12)

4.2.3.5 CU28

Caso de uso: El sistema debe mostrarle al administrativo o al monitor una lista con los datos de contacto de los senderistas que estaban apuntados a la ruta que se ha eliminado

Objetivo: Informar a los senderistas de la cancelación de una ruta

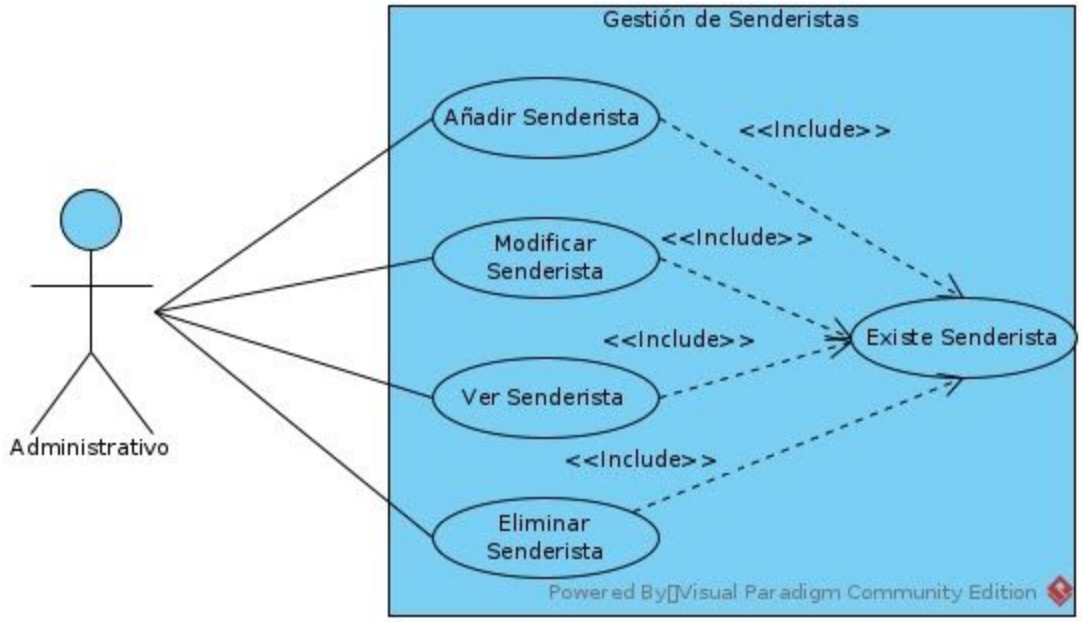
Contexto: El senderista debe existir en el sistema

Actor principal: Sistema

Escenario principal:

1. El administrativo o el monitor deben borrar una ruta
2. El sistema debe mostrar los datos de los senderistas que estaban apuntados a dicha actividad

4.2.4 Gestión de senderistas



4.2.4.1 CU16

Caso de uso: El administrativo quiere añadir un senderista al sistema

Objetivo: Administrar los senderistas de forma que se pueda añadir información al sistema

Contexto: El senderista no debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir el DNI del senderista
2. El sistema debe almacenar la nueva información

Extensiones:

1. a El sistema debe comprobar que el senderista no exista(CU17)

4.2.4.2 CU17

Caso de uso: Comprobar si un senderista está en el sistema

Objetivo: Evitar la repetición de la información en el sistema

Contexto: Debe de existir como mínimo un senderista en el sistema

Actor principal: Sistema

Escenario principal:

1. El administrativo debe introducir el DNI del senderista cuya existencia queremos comprobar
2. El sistema debe comparar la información introducida con la información disponible
3. El sistema debe devolver TRUE si el senderista se encuentra en el sistema y FALSE si no se encuentra

4.2.4.3 CU18

Caso de uso: El administrativo quiere modificar la información de un senderista del sistema

Objetivo: Administrar los senderistas de forma que se pueda actualizar la información del sistema

Contexto: El senderista debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir el DNI del senderista
2. El administrativo debe modificar los datos del senderista que desee
3. El sistema debe almacenar la nueva información

Extensiones:

1. a El sistema debe comprobar que el senderista exista(CU17)

4.2.4.4 CU19

Caso de uso: El administrativo quiere eliminar un senderista del sistema

Objetivo: Administrar los senderistas de forma que se pueda eliminar información del sistema

Contexto: El senderista debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir el DNI del senderista
2. El sistema debe eliminar la información del senderista

Extensiones:

1. a El sistema debe comprobar que el senderista exista(CU17)

4.2.4.5 CU20

Caso de uso: El administrativo quiere ver la información de un senderista del sistema

Objetivo: Administrar los senderistas de forma que se pueda visualizar la información del sistema

Contexto: El senderista debe de existir en el sistema

Actor principal: Administrativo

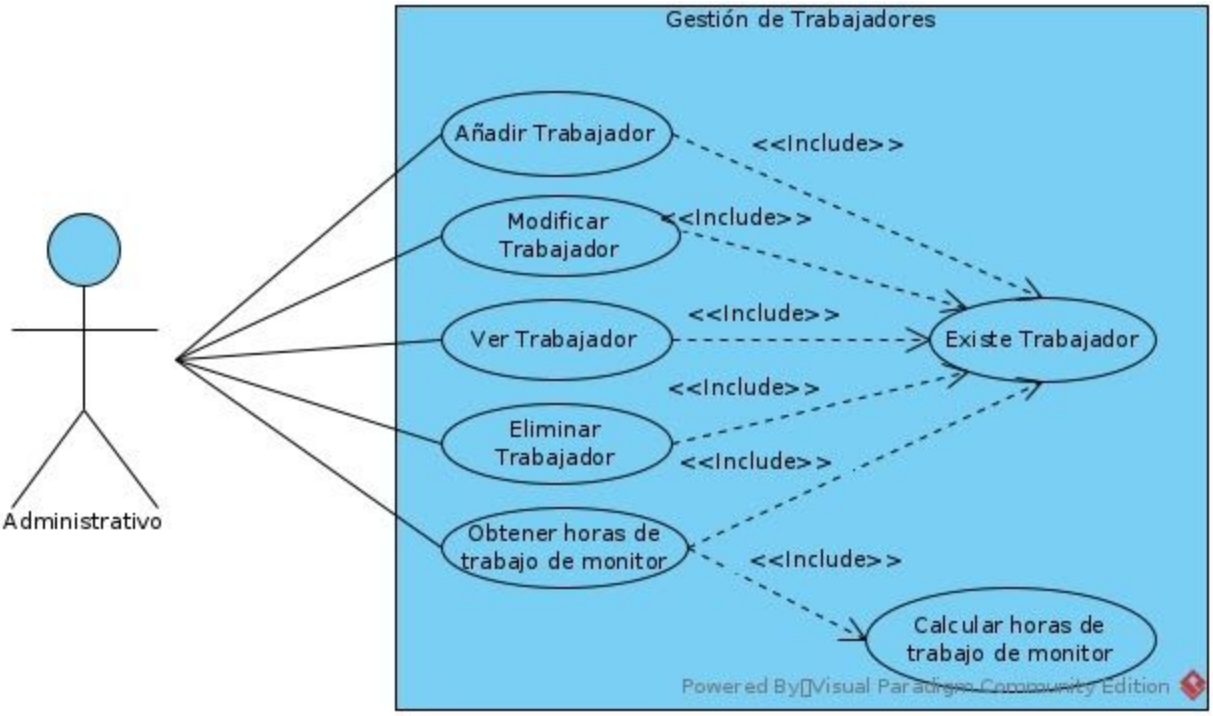
Escenario principal:

1. El administrativo debe introducir el DNI del senderista
2. El sistema debe mostrar la información

Extensiones:

1. a El sistema debe comprobar que el senderista exista(CU17)

4.2.5 Gestión de trabajadores



4.2.5.1 CU21

Caso de uso: El administrativo quiere añadir un trabajador al sistema

Objetivo: Administrar los trabajadores de forma que se pueda añadir información al sistema

Contexto: El trabajador no debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir el DNI del trabajador
2. El sistema debe almacenar la nueva información

Extensiones:

1. a El sistema debe comprobar que el trabajador no exista(CU22)

4.2.5.2 CU22

Caso de uso: Comprobar si un trabajador está en el sistema

Objetivo: Evitar la repetición de la información en el sistema

Contexto: Debe de existir como mínimo un trabajador en el sistema

Actor principal: Sistema

Escenario principal:

1. El administrativo debe introducir el DNI del trabajador cuya existencia queremos comprobar
2. El sistema debe comparar la información introducida con la información disponible
3. El sistema debe devolver TRUE si el trabajador se encuentra en el sistema y FALSE si no se encuentra

4.2.5.3 CU23

Caso de uso: El administrativo quiere modificar la información de un trabajador del sistema

Objetivo: Administrar los trabajadores de forma que se pueda actualizar la información del sistema

Contexto: El trabajador debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir el DNI del trabajador
2. El administrativo debe modificar los datos del trabajador que desee
3. El sistema debe almacenar la nueva información

Extensiones:

1. a El sistema debe comprobar que el trabajador exista(CU22)

4.2.5.4 CU24

Caso de uso: El administrativo quiere eliminar un trabajador del sistema

Objetivo: Administrar los trabajadores de forma que se pueda eliminar información del sistema

Contexto: El trabajador debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir el DNI del trabajador
2. El sistema debe eliminar la información del trabajdor

Extensiones:

1. a El sistema debe comprobar que el trabajador exista(CU22)

4.2.5.5 CU25

Caso de uso: El administrativo quiere ver la información de un trabajador del sistema

Objetivo: Administrar los trabajadores de forma que se pueda visualizar la información del sistema

Contexto: El trabajador debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir el DNI del trabajador
2. El sistema debe mostrar la información

Extensiones:

1. a El sistema debe comprobar que el trabajador exista(CU22)

4.2.5.6 CU26

Caso de uso: El administrativo quiere comprobar las horas de trabajo de los monitores

Objetivo: Administrar los monitores de tal forma que se puedan visualizar sus horas de trabajo en el sistema

Contexto: El monitor debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir la información del monitor
2. El sistema debe mostrar las horas trabajadas

Extensiones:

1. a El sistema debe comprobar que el monitor exista(CU22)

4.2.5.7 CU27

Caso de uso: El administrativo quiere poder pasar la información de horas de trabajo de un monitor al sistema de nóminas

Objetivo: Conectar las horas trabajadas de cada monitor con el sistema de nóminas de estos

Contexto: El monitor debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir la información del monitor
2. El sistema debe pasar las horas trabajadas del monitor introducido al sistema de nóminas

Extensiones:

1. a El sistema debe comprobar que el monitor exista(CU22)