

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:

- Añadirles al tablón haciendo uso de las funcionalidad de historias de usuario
- Añadirles 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

Todos los documentos citados en los párrafos anteriores así como las tarjetas con las historias de usuario, se encuentran respectivamente en la sección Knowledge Base y en el Panel de Desarrollo de la sección Agile Boards de nuestra instancia de YouTrack cuyo enlace es el siguiente:

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

3. Especificación de requisitos (P2)

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 permitir a los administrativos gestionar la información de los parques naturales de Andalucía abiertos a los visitantes, así como sus senderos y rutas. Las rutas serán dinámicas es decir se crean con una fecha y un recorrido determinado. En caso de que se elimine una ruta debe suministrarse a los administrativos la información de los senderistas que estaban apuntados a ella para que les avisen de que se ha cancelado. También deben poder crearse rutas exclusivas. Por ejemplo: para los estudiantes de un colegio. O adaptadas a las necesidades especiales de los senderistas

RF2. Senderistas y rutas

El sistema debe permitir al administrador la reserva y la cancelación de una ruta por parte de un senderista, siempre y cuando queden más de 15 minutos para el inicio de la misma. Si esta ruta quedará llena, no se podrían apuntar más senderistas a ella. Por otro lado, como al eliminar una reserva quedará en la ruta una plaza libre, el sistema debe permitir que otro senderista se pueda apuntar a esta ruta dentro del tiempo mencionado anteriormente.

RF3. Información de rutas

Previo al inicio de cada ruta el sistema debe suministrar al monitor la información de la ruta que va a realizar así como el número de senderistas que van a participar e información sobre los mismos por si tuviesen requisitos especiales ej. movilidad reducida. También el sistema deberá ser capaz de suministrar la información a un segundo monitor suplente, el cual se ocupará de la ruta en caso de que el monitor principal sea incapaz de realizarla.

RF4. Notificación de incidencias

El sistema debe permitir a los monitores notificar incidencias que ocurran durante las rutas. Estas notificaciones las recibirán los administrativos, para que así estos puedan por ejemplo, eliminar o modificar una ruta o sendero del parque de forma temporal o actualizar la información de dicha ruta.

RF5. Gestión de nóminas

El sistema debe permitir a los administrativos generar un reporte de las horas trabajadas por cada monitor, en base al número de rutas que este ha realizado y la duración de las mismas.

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 (P2)

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 la información de los parques PARA actualizar la información del sistema

Descripción:

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

Haciendo uso del identificador del parque, el sistema primeramente deberá comprobar si existe o no:

- Si no existe, el administrativo lo añadirá o cambiará el identificador introducido.
- Si existe, el administrador elegirá qué hacer con el parque: modificarlo, eliminarlo o simplemente consultarlo.

Criterios de aceptación:

1. El sistema debe comprobar que sea un administrativo el que modifica la información del parque ya que son los únicos con autoridad para hacerlo
2. El sistema debe impedir registrar más de una vez el mismo parque
3. El sistema debe impedir que se registre un parque sin rellenar los campos obligatorios del formulario
4. Si se elimina un parque del sistema este debe encargarse de eliminar también los senderos y las rutas asociadas al mismo

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

Descripción:

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

Haciendo uso del identificador del parque, el sistema primeramente deberá comprobar si existe o no:

- Si no existe, el administrativo deberá introducir un identificador válido para buscar el sendero a tratar.
- Si existe, el administrador introducirá el identificador del sendero y el sistema deberá comprobar si existe o no:
 - Si no existe, el administrativo lo añadirá o cambiará el identificador introducido.
 - Si existe, el administrador elegirá que hacer con el sendero: modificarlo, eliminarlo o simplemente consultarlo.

Criterios de aceptación:

1. El sistema debe comprobar que sea un administrativo el que modifica la información del sendero ya que son los únicos con autoridad para hacerlo.
2. El sistema debe impedir registrar más de una vez el mismo sendero.
3. El sistema debe impedir que se registre un sendero sin rellenar los campos obligatorios del formulario.
4. Si se elimina un sendero del sistema este debe encargarse de eliminar también las rutas asociadas al mismo

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

Descripción:

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

Haciendo uso del identificador del parque, el sistema primeramente deberá comprobar si existe o no:

- Si no existe, el administrativo deberá introducir un identificador válido para buscar la ruta a tratar.
- Si existe, el administrador introducirá el identificador de la ruta y el sistema deberá comprobar si existe o no:
 - Si no existe, el administrativo la añadirá o cambiará el identificador introducido.
 - Si existe, el administrador elegirá qué hacer con la ruta: modificarla, eliminarla o simplemente consultarla.

Criterios de aceptación:

1. El sistema debe impedir registrar más de una ruta a la vez en el mismo sendero y a la misma hora.
2. El sistema debe impedir que se registre una ruta sin rellenar los campos obligatorios del formulario.
3. Si se elimina una ruta del sistema este debe encargarse de notificarlo a los administrativos para que avisen a los senderistas

HU4. COMO monitor QUIERO administrar la información de las rutas PARA actualizar la información del sistema

Descripción:

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

Haciendo uso del identificador del parque, el sistema primeramente deberá comprobar si existe o no:

- Si no existe, el monitor deberá introducir un identificador válido para buscar la ruta a tratar.
- Si existe, el monitor introducirá el identificador de la ruta y el sistema deberá comprobar si existe o no:
 - Si no existe, el monitor la añadirá o cambiará el identificador introducido.
 - Si existe, el monitor elegirá qué hacer con la ruta: modificarla, eliminarla o simplemente consultarla.

Criterios de aceptación:

1. El sistema debe impedir registrar más de una ruta a la vez en el mismo sendero y a la misma hora.
2. El sistema debe impedir que se registre una ruta sin rellenar los campos obligatorios del formulario.
3. Si se elimina una ruta del sistema este debe encargarse de notificarlo a los administrativos para que avisen a los senderistas

HU5. COMO administrativo QUIERO administrar la información de los senderistas PARA actualizar la información del sistema

Descripción:

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

Haciendo uso del DNI del senderista, el sistema primeramente deberá comprobar si existe o no:

- Si no existe, el administrativo lo añadirá o cambiará el DNI introducido.
- Si existe, el administrador elegirá qué hacer con el senderista: modificarlo, eliminarlo o simplemente consultarlo.

Criterios de aceptación:

1. El sistema debe comprobar que sea un administrativo el que modifica la información almacenada sobre un senderista ya que son los únicos con autoridad para hacerlo.
2. El sistema debe impedir registrar más de una vez el mismo senderista.
3. El sistema debe impedir que se registre un senderista sin rellenar los campos obligatorios del formulario.

HU6. COMO administrativo QUIERO añadir y eliminar senderistas a las rutas PARA poder organizarlas

Descripción:

Los administrativos deben poder apuntar y desapuntar a los senderistas a las distintas rutas del parque.

Una vez el administrativo acceda a la ruta (como se explica en la HU3), haciendo uso del DNI del senderista, los administrativos podrán añadir y eliminar senderistas a la ruta.

Criterios de aceptación:

1. El sistema debe comprobar que sea un administrativo el que apunta o desapunta un senderista de la ruta ya que son los únicos con autoridad para hacerlo.
2. El sistema debe impedir registrar más de una vez el mismo senderista.
3. El sistema debe impedir que se registre un senderista sin rellenar los campos obligatorios del formulario.

HU7. COMO monitor QUIERO disponer de un listado de los senderistas que participan en la ruta PARA poder prepararla

Descripción:

Los monitores deben poder acceder a un listado con la información necesaria sobre la ruta que van a realizar.

Una vez el monitor acceda a la ruta (como se explica en la HU4), estos seleccionarán la opción de mostrar senderistas para acceder al listado de estos, y así poder preparar la ruta.

Criterios de aceptación:

1. El sistema debe comprobar que sea un monitor el que solicita este listado ya que son los únicos con autoridad para hacerlo.
2. El listado debe incluir la siguiente información: Senderos por los que pasa la ruta, duración de la ruta, número de senderistas que participan en la ruta y si alguno tiene necesidades especiales, así como el material necesario para la realización de la misma.
3. El sistema debe impedir que se extraiga este listado hasta que no falten menos de 15 minutos para que empiece dicha ruta ya que hasta entonces la información del mismo puede cambiar .

HU8. COMO monitor QUIERO poder notificar incidencias en las rutas PARA informar a los administrativos del parque de lo ocurrido

Descripción:

Los monitores deben poder notificar incidencias sobre las rutas.

Una vez el monitor acceda a la ruta (como se explica en la HU4), los monitores podrán dar parte sobre las incidencias que hayan ocurrido en esta para que estas se almacenen en el sistema.

Los administrativos, accediendo a la ruta (como se explica en la HU3), podrán consultarla y así observar las incidencias que hayan ocurrido.

Criterios de aceptación:

1. El sistema debe comprobar que sea un monitor y que dicho monitor haya realizado esa ruta para poder notificar las incidencias.
2. El sistema debe impedir que se registre una incidencia sin rellenar los campos obligatorios del formulario.
3. El sistema debe notificar a los administrativos de la incidencia para que actúen en consecuencia.

HU9. COMO administrativo QUIERO administrar la información de la plantilla PARA actualizar la información del sistema

Descripción:

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

Haciendo uso del DNI del trabajador, el sistema primeramente deberá comprobar si existe o no:

- Si no existe, el administrativo lo añadirá o cambiará el DNI introducido.
- Si existe, el administrador elegirá qué hacer con el trabajador: modificarlo, eliminarlo o simplemente consultarlo.

Criterios de aceptación:

1. El sistema debe comprobar que sea un administrativo el que añade un trabajador ya que son los únicos con autoridad para hacerlo.
2. El sistema debe impedir registrar más de una vez el mismo trabajador.
3. El sistema debe impedir que se registre un trabajador sin rellenar los campos obligatorios del formulario.

HU10. COMO administrativo QUIERO disponer del cómputo de horas trabajadas de los monitores PARA poder pagarles

Descripción:

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

Haciendo uso del DNI del trabajador, el sistema primeramente deberá comprobar si existe o no:

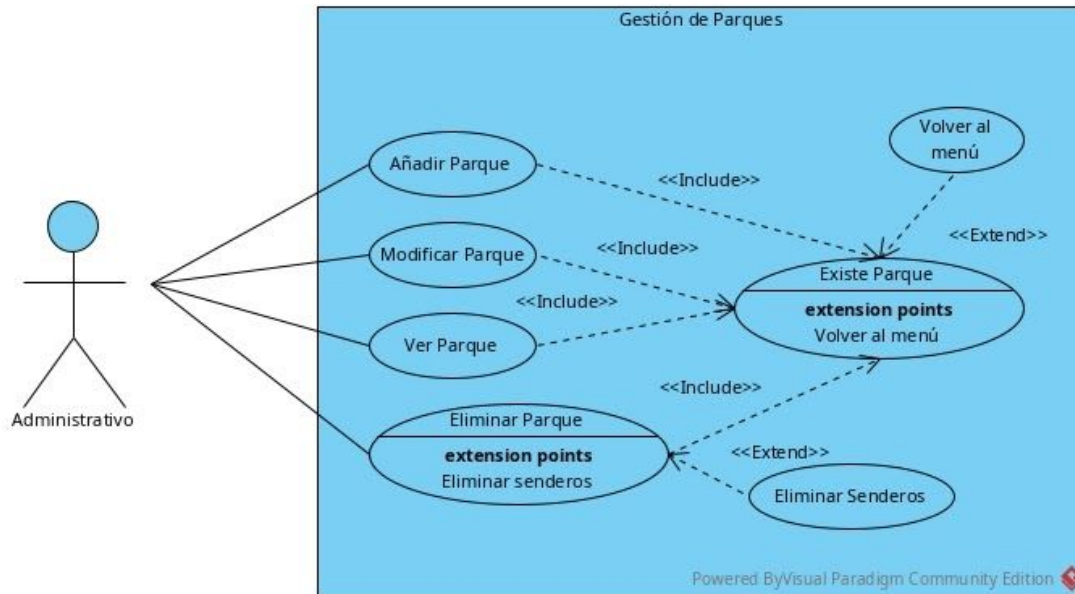
- Si no existe, el administrativo cambiará el DNI introducido para buscar un monitor que se encuentre en el sistema.
- Si existe, el administrador indicará al sistema que le muestre el cómputo de horas trabajadas del monitor correspondiente.

Criterios de aceptación:

1. El sistema debe comprobar que sea un administrativo el que consulte esta información ya que son los únicos con autoridad para hacerlo.
2. En un futuro este reporte se integrará automáticamente con el sistema de nóminas por lo que el formato de salida debe de ser compatible con este sistema.

4.2 Casos de uso

4.2.1 Gestión de parques



“La figura 1 muestra el diagrama de casos de uso para la gestión de los 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 el identificador del parque
2. El sistema debe comprobar que el parque no exista(CU2)
3. El administrativo debe continuar rellenando el resto del formulario y el sistema debe almacenar la nueva información

Extensiones:

1. Si el parque existe en el sistema, la operación se cancela y vuelve al menú principal

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. Se introduce el identificador del parque cuya existencia queremos comprobar
2. El sistema debe comparar el identificador introducido con los identificadores que existen en el sistema
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 el identificador del parque
2. El sistema debe comprobar que el parque exista(CU2)
3. El administrativo debe continuar rellenando el resto del formulario y el sistema debe almacenar la nueva información

Extensiones:

1. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

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 el identificador del parque
2. El sistema debe comprobar que el parque exista(CU2)
3. El sistema, este debe eliminar la información del parque.

Extensiones:

1. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal
2. 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 para comprobar la información almacenada sobre el mismo (superficie, premios...) y el estado en el que se encuentran sus senderos y rutas.

Objetivo: Consultar la información de un parque almacenado en el sistema

Contexto: El parque debe de existir en el sistema

Actor principal: Administrativo

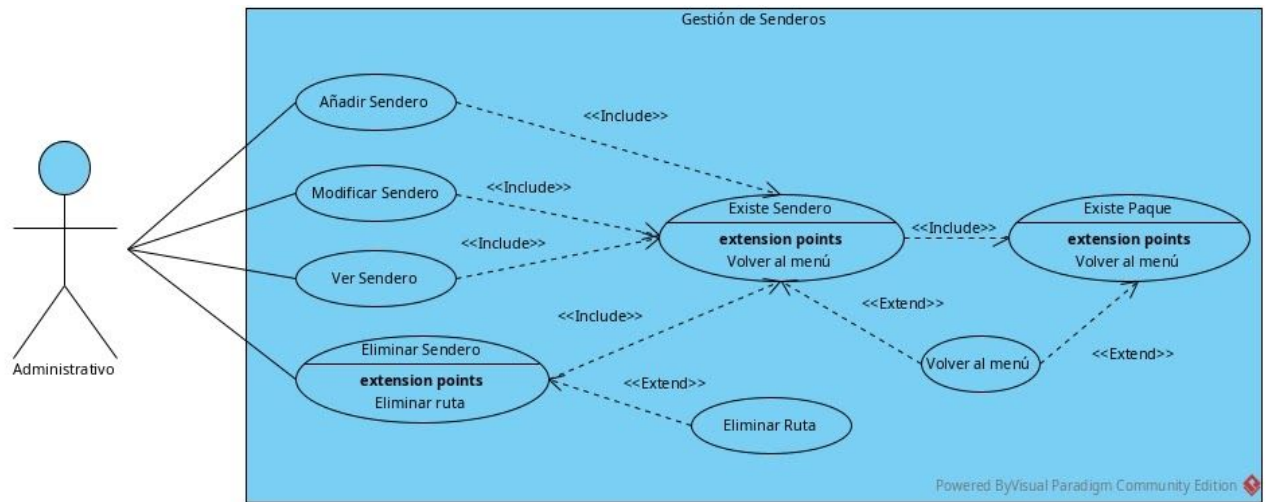
Escenario principal:

1. El administrativo debe introducir el identificador del parque
2. El sistema debe comprobar que el parque exista(CU2)
3. El sistema se imprime por pantalla la información

Extensiones:

1. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

4.2.2 Gestión de senderos



“La figura 2 muestra el diagrama de casos de uso para la gestión de los 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 el identificador del parque al que se desea añadir y el identificador del sendero
2. El sistema debe comprobar que el sendero no exista en ese parque (CU7)
3. El administrativo debe continuar rellenando el resto del formulario y el sistema debe almacenar la nueva información

Extensiones:

1. Si el sendero existe en el parque, la operación se cancela y vuelve al menú principal
2. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

4.2.2.2 CU7

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

Objetivo: Validar la integridad de la información del sistema

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

Actor principal: Sistema

Escenario principal:

1. Se introduce el identificador del sendero cuya existencia queremos comprobar y el identificador del parque al que pertenece
2. El sistema debe comprobar que el parque exista(CU2)
3. El sistema debe comparar el identificador del sendero introducido con los identificadores de senderos que existen en el parque
4. El sistema debe devolver TRUE si el sendero se encuentra en el 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 el identificador del parque en el que se encuentra el sendero y el identificador del sendero que se desea modificar
2. El sistema debe comprobar que el sendero exista en ese parque (CU7)
3. El administrativo debe continuar rellenando el resto del formulario y el sistema debe almacenar la nueva información

Extensiones:

1. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
2. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

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 el identificador del parque en el que se encuentra el sendero y el identificador del sendero que se desea eliminar
2. El sistema debe comprobar que el sendero exista en ese parque (CU7)
3. Si el sendero existe en el parque el sistema debe eliminar la información del mismo

Extensiones:

1. El sistema debe borrar la información de las rutas asociadas a dicho sendero(CU14)
2. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
3. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

4.2.2.5 CU10

Caso de uso: El administrativo quiere ver la información de un sendero del sistema para comprobar la información almacenada sobre el mismo y el estado en el que se encuentra.

Objetivo: Consultar la información de un sendero almacenado en el sistema

Contexto: El sendero debe de existir en el sistema

Actor principal: Administrativo

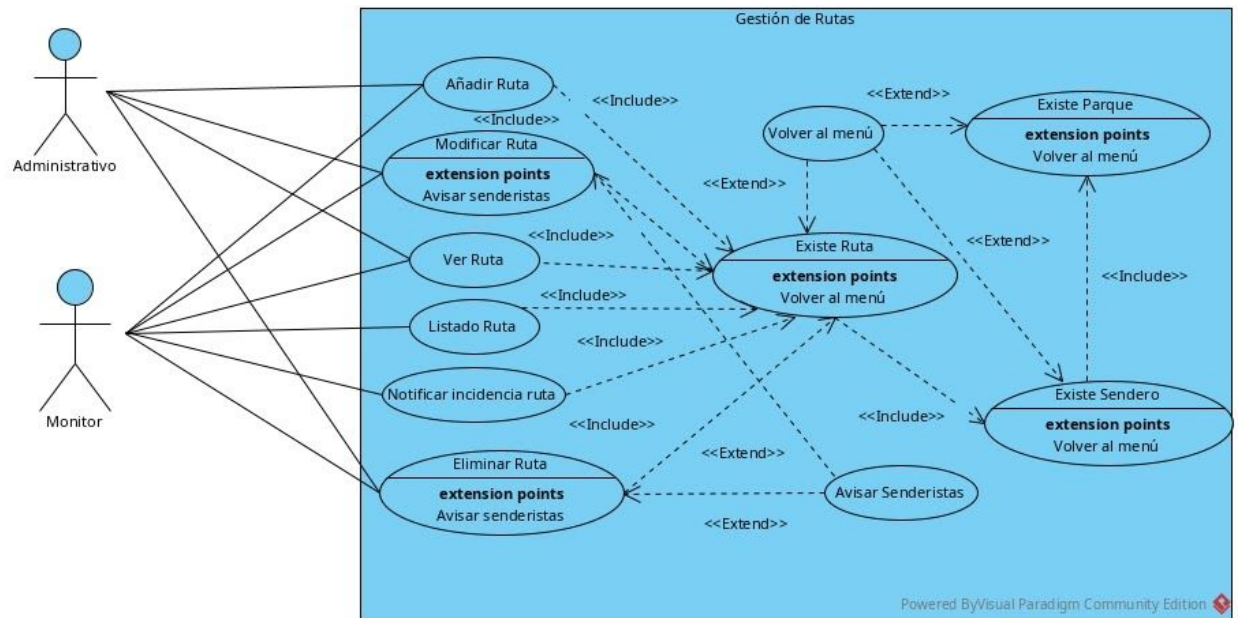
Escenario principal:

1. El administrativo debe introducir el identificador del parque en el que se encuentra el sendero y el identificador del sendero que se desea mostrar
2. El sistema debe comprobar que el sendero exista en ese parque (CU7)
3. Si el sendero existe en el parque el sistema debe mostrar la información del mismo

Extensiones

1. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
2. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

4.2.3 Gestión de rutas



"La figura 3 muestra el diagrama de casos de uso para la gestión de las 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 monitor debe introducir la fecha, la hora, el identificador del sendero y el identificador del parque al que pertenece la ruta
2. El sistema debe comprobar que no exista una ruta en la misma fecha a la misma hora en ese sendero de ese parque(CU12)
3. El administrativo o monitor debe continuar rellenando el resto del formulario y el sistema debe almacenar la nueva información

Extensiones

1. Si existe, la operación se cancela y vuelve al menú principal
2. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
3. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

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. Se introduce el identificador del sendero y el identificador del parque al que pertenece la ruta
2. El sistema debe comprobar que el parque exista(CU2)
3. El sistema debe comparar el identificador del sendero introducido con los identificadores de senderos que existen en el parque
4. El sistema debe comprobar la fecha y la hora de las rutas asociadas a ese sendero
5. El sistema debe devolver TRUE si existe dicha ruta y false si no existe

Extensiones:

1. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
2. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

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 la fecha, la hora, el identificador del sendero y el identificador del parque al que pertenece la ruta
2. El sistema debe comprobar que exista una ruta en la misma fecha a la misma hora en ese sendero de ese parque(CU12)
3. El administrativo o monitor debe continuar rellenando el resto del formulario y el sistema debe almacenar la nueva información

Extensiones:

1. El sistema debe recordarle al administrador o monitor avisar a los excursionistas apuntados a dicha ruta(CU15)
3. Si no existe, la operación se cancela y vuelve al menú principal
4. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
5. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

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 la fecha, la hora, el identificador del sendero y el identificador del parque al que pertenece la ruta
2. El sistema debe comprobar que exista una ruta en la misma fecha a la misma hora en ese sendero de ese parque(CU12)
3. El sistema elimina la información de dicha ruta del parque y la eliminará de la lista de rutas del monitor y de la listas de rutas de los senderistas que estuviesen apuntados.

Extensiones:

1. El sistema debe recordarle al administrador o monitor avisar a los excursionistas apuntados a dicha ruta(CU15)
2. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal
3. Si la ruta no existe, la operación se cancela y vuelve al menú principal
4. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal

4.2.3.5 CU15

Caso de uso: El sistema debe mostrarle al administrativo o 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 sistema debe mostrar los datos de contacto los senderistas que estaban apuntados a dicha actividad

4.2.3.6 CU16

Caso de uso: El administrativo o el monitor quieren poder ver la información de una ruta del sistema para preparar la actividad y adaptarla a los requisitos de los senderistas.

Objetivo: Consultar la información de una ruta almacenada en el sistema para poder preparar las rutas adecuadamente

Contexto: La ruta debe de existir en el sistema

Actor principal: Administrativo o monitor

Escenario principal:

1. El administrativo o monitor debe introducir la fecha, la hora, el identificador del
2. sendero y el identificador del parque al que pertenece la ruta
3. El sistema debe comprobar que exista una ruta en la misma fecha a la misma
4. hora en ese sendero de ese parque(CU12)
5. El sistema muestra la información de dicha ruta del sistema.

Extensiones:

1. Si no existe, la operación se cancela y vuelve al menú principal
2. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
3. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

4.2.3.7 CU30

Caso de uso: El monitor quiere un listado con la información de los senderistas que participan en la ruta para tener información sobre el grupo que va a realizar la actividad

Objetivo: Preparar el material y adaptar la ruta a las necesidades de los senderistas

Contexto: La ruta debe existir en el sistema

Actor principal: Monitor

Escenario principal:

1. El monitor debe introducir la fecha, la hora, el identificador del sendero y el
2. identificador del parque al que pertenece la ruta
3. El sistema debe comprobar que exista una ruta en la misma fecha a la misma
4. hora en ese sendero de ese parque(CU12)
5. El sistema debe generar un documento que incluya la información de la ruta y la de los senderistas que van a participar en ella

Extensiones:

1. Si no existe, la operación se cancela y vuelve al menú principal
2. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
3. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

4.2.3.8 CU31

Caso de uso: El monitor quiere poder notificar a los administrativos de lo ocurrido durante la ruta

Objetivo: Dar parte a los administrativos de lo ocurrido

Contexto: La ruta debe existir en el sistema

Actor principal: Monitor

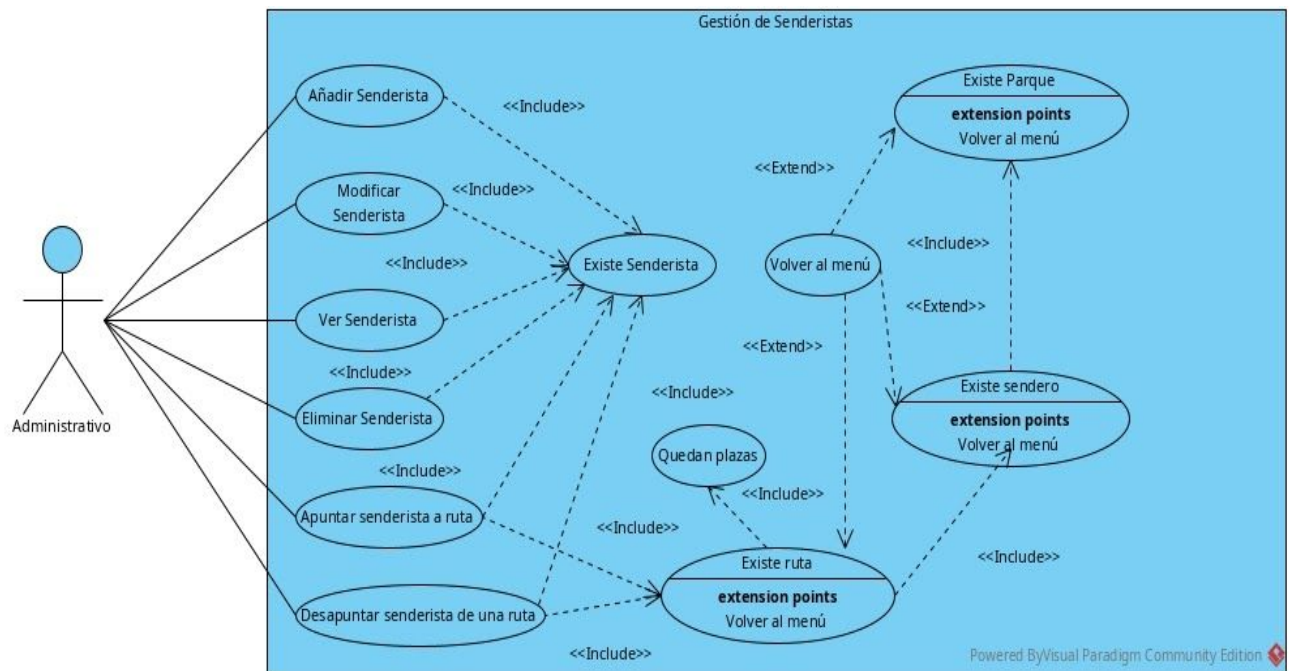
Escenario principal:

1. El monitor debe introducir la fecha, la hora, el identificador del sendero y el
2. identificador del parque al que pertenece la ruta
3. El sistema debe comprobar que exista una ruta en la misma fecha a la misma
4. hora en ese sendero de ese parque(CU12)
5. El sistema debe crear un parte de incidencias que el monitor deberá
6. rellenar y que el sistema hará llegar a los administrativos

Extensiones:

1. Si no existe, la operación se cancela y vuelve al menú principal
2. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
3. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal.

4.2.4 Gestión de senderistas



“La figura 4 muestra el diagrama de casos de uso para la gestión de senderistas”

4.2.4.1 CU17

Caso de uso: El administrativo quiere dar de alta un senderista en el 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 comprobar que el senderista no exista(CU18)
3. El administrativo debe continuar rellenando el resto del formulario y el sistema debe almacenar la nueva información

Extensiones:

1. Si existe la operación se cancela y se vuelve al menú principal

4.2.4.2 CU18

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. Se introduce 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 CU19

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

Objetivo: Corregir la información de los senderistas en caso de que se hayan equivocado al introducirla o haya cambiado algo. Ej: su teléfono de contacto.

Contexto: El senderista debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

- El administrativo debe introducir el DNI del senderista
- El sistema debe comprobar que el senderista exista(CU18)
- El administrativo debe continuar rellenando el resto del formulario y el sistema debe almacenar la nueva información

Extensiones:

1. Si no existe la operación se cancela y se vuelve al menú principal

4.2.4.4 CU20

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 comprobar que el senderista exista(CU18)
3. El sistema debe eliminar la información del senderista y borrarlo de las rutas en las que este participase

Extensiones:

1. Si no existe el senderista la operación se cancela y se vuelve al menú principal

4.2.4.5 CU21

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

Objetivo: Poder ver la información que el sistema almacena del senderista por si hay que contactar con el.

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 comprobar que el senderista exista(CU18)
3. El sistema debe mostrar la información del senderista

Extensiones:

1. Si no existe la operación se cancela y se vuelve al menú principal

4.2.4.6 CU22

Caso de uso: El administrativo quiere apuntar un senderista a una ruta

Objetivo: Añadir un nuevo participante a una ruta

Contexto: El senderista debe de existir en el sistema y deben quedar plazas en la ruta

Actor principal: Administrativo

Escenario principal:

1. El administrativo o monitor debe introducir la fecha, la hora, el identificador del sendero y el identificador del parque al que pertenece la ruta
2. El sistema debe comprobar que exista una ruta en la misma fecha a la misma hora en ese sendero de ese parque(CU12)
3. El sistema debe comprobar si quedan plazas disponibles(CU32)
4. El administrativo debe introducir el DNI del senderista
5. El sistema debe comprobar que el senderista exista(CU18)
6. El administrativo debe continuar rellenando el formulario de inscripción y el sistema debe almacenar la nueva información además de apuntar el senderista a la ruta

Extensiones:

1. Si existe el senderista se añade a la ruta
2. Si no quedan plazas la operación se cancela y vuelve al menú principal
3. Si no existe, la operación se cancela y vuelve al menú principal
4. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
5. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

4.2.4.7 CU23

Caso de uso: El administrativo quiere eliminar un senderista de una ruta

Objetivo: Dar de baja un participante de una ruta ya que este no va a poder asistir y dejando su plaza libre para otro senderista

Contexto: El senderista debe de existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo o monitor debe introducir la fecha, la hora, el identificador del sendero y el identificador del parque al que pertenece la ruta
2. El sistema debe comprobar que exista una ruta en la misma fecha a la misma hora en ese sendero de ese parque(CU12)
3. El administrativo debe introducir el DNI del senderista
4. El sistema debe comprobar que el senderista exista(CU18)
5. El sistema debe borrarlo de la lista de senderistas de esa ruta

Extensiones:

1. Si no existe el senderista la operación se cancela y vuelve al menú principal
2. Si no existe, la operación se cancela y vuelve al menú principal
3. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
4. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

4.2.4.8 CU32

Caso de uso: Comprobar si quedan plazas en la ruta

Objetivo: Gestionar el aforo de las rutas

Contexto: La ruta debe existir en el sistema

Actor principal: Sistema

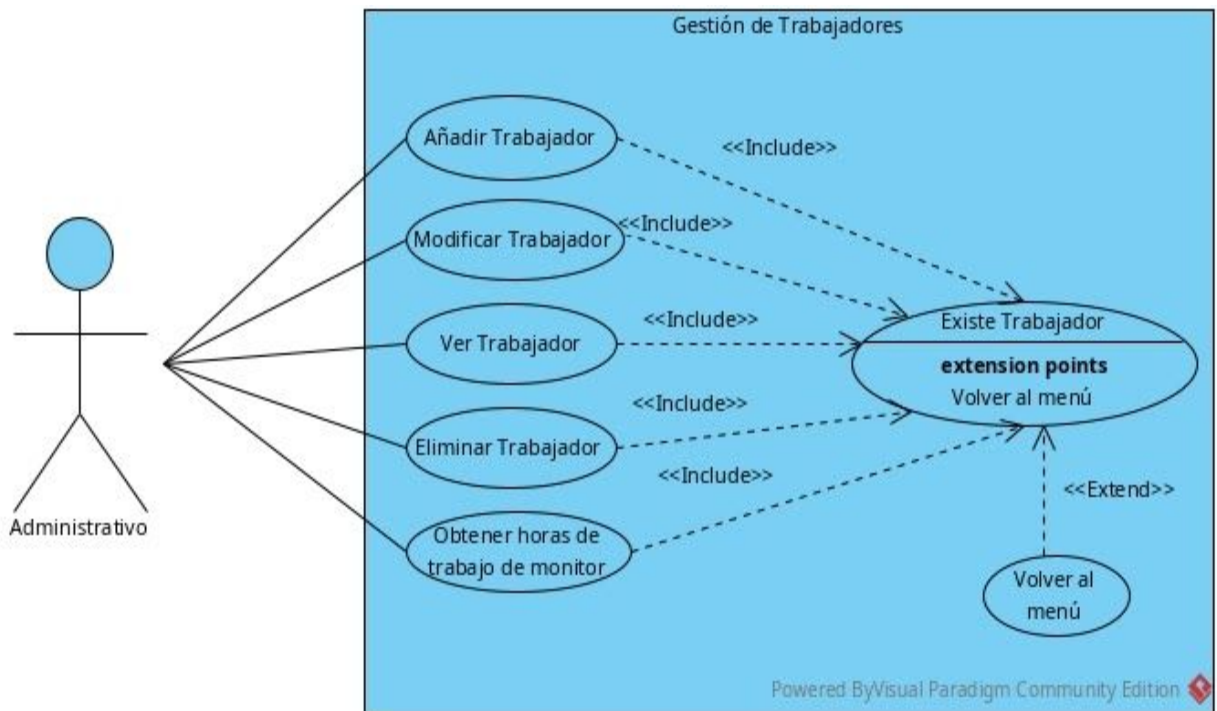
Escenario principal:

1. Se introduce la fecha, la hora, el identificador del sendero y el identificador del
2. parque al que pertenece la ruta
3. El sistema debe comprobar que exista una ruta en la misma fecha a la misma
4. hora en ese sendero de ese parque(CU12)
5. El sistema devolverá TRUE si hay al menos una plaza libre o FALSE si no
6. la hay

Extensiones:

1. Si no existe, la operación se cancela y vuelve al menú principal
2. Si el sendero no existe en el parque, la operación se cancela y vuelve al menú principal
3. Si el parque no existe en el sistema, la operación se cancela y vuelve al menú principal

4.2.5 Gestión de trabajadores



“La figura 5 muestra el diagrama de casos de uso para la gestión de trabajadores”

4.2.5.1 CU24

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

Objetivo: Dar de alta un nuevo trabajador del 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 comprobar que el trabajador no exista(CU25)
3. El administrativo debe continuar rellenando el resto del formulario y el sistema debe almacenar la nueva información

Extensiones:

1. Si existe la operación se cancela y se vuelve al menú principal

4.2.5.2 CU25

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. Se introduce 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 CU26

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

Objetivo: Editar y corregir la información que el sistema almacena sobre los trabajadores para poder corregirla en caso de error

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 comprobar que el trabajador exista(CU25)
3. El administrativo debe continuar rellenando el resto del formulario y el sistema debe almacenar la nueva información

Extensiones:

1. Si no existe la operación se cancela y se vuelve al menú principal

4.2.5.4 CU27

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

Objetivo: Eliminar del sistema un extrabajador de la empresa

Contexto: El trabajador debe existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir el DNI del trabajador
2. El sistema debe comprobar que el trabajador exista(CU25)
3. El sistema debe eliminar la información del trabajador y si es un monitor, borrarlo de todas las rutas en las que sea el encargado.

Extensiones:

1. Si no existe la operación se cancela y se vuelve al menú principal

4.2.5.5 CU28

Caso de uso: El administrativo quiere ver la información de un trabajador del sistema para comprobar la información almacenada sobre el mismo y sus datos de contacto.

Objetivo: Consultar la información de un trabajador almacenada en el sistema

Contexto: El trabajador debe existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El administrativo debe introducir el DNI del trabajador
2. El sistema debe comprobar que el trabajador exista(CU25)
3. El sistema debe mostrar su información

Extensiones:

1. Si no existe la operación se cancela y se vuelve al menú principal

4.2.5.6 CU29

Caso de uso: El administrativo quiere obtener el cómputo de 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 existir en el sistema

Actor principal: Administrativo

Escenario principal:

1. El sistema debe mostrar un listado con las horas trabajadas de cada monitor y su DNI

5. Referencias

La documentación utilizada para elaborar esta práctica ha sido la aportada por la profesora en el moodle de la asignatura. Esto incluye los contenidos teóricos aportados en los diferentes documentos de cada una de las semanas de prácticas y la información adjuntada en los siguientes enlaces:

- <https://medium.com/@wc.testing.qa/agile-testing-de-historias-de-usuario-criterios-de-aceptaci%C3%B3n-y-c%C3%B3mo-escribir-una-historia-2a94149ecec7>
- <https://apiumhub.com/es/tech-blog-barcelona/como-escribir-buenas-historias-de-usuario/>
- <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>