

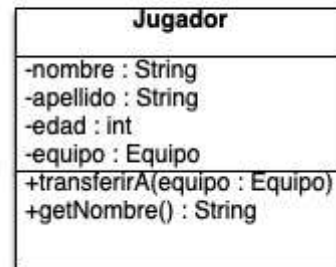
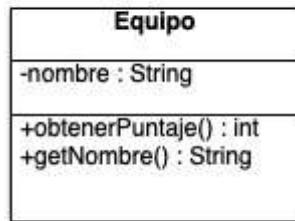
Pregunta 1

Finalizado

Sin calificar

Teniendo en cuenta las siguientes clases y la información que se suministra a continuación, indique qué tipo de relación existe entre las clases.

Información: Se está desarrollando un software para mantener la información de la liga Qatari profesional de water polo. Según nos explica la persona a cargo, cada equipo está compuesto por 18 jugadores de entre 18 y 39 años. Cada jugador solo puede ser parte de un equipo a la vez. Debido a la inestabilidad del barril de crudo, en algunas ocasiones, los equipos quiebran por lo que sus jugadores son transferidos a otros equipos.



Seleccione una:

- ☐ a. Realización
- ☐ b. Asociación
- ☐ c. Dependencia
- ☐ d. Generalización
- ☐ e. Composición
- ☒ f. Agregación

Respuesta correcta

La respuesta correcta es: Agregación

Pregunta 2

Finalizado

Sin calificar

Una vez que se selecciona un conjunto de PBIs para un Sprint, estas PBIs no pueden ser modificadas por el equipo de desarrollo (por ejemplo, agregarle detalles) durante el Sprint

Seleccione una:

- ☐ Verdadero
- ☒ Falso

La respuesta correcta es 'Falso'

Pregunta 3

Finalizado

Sin calificar

Suponga que estamos a punto de comenzar el sprint #2 de un proyecto.

Indique si la siguiente frase es verdadera o falsa:

En este punto todos los elementos del product backlog tienen el mismo nivel de detalle

Seleccione una:

☐ Verdadero

☒ Falso

La respuesta correcta es 'Falso'

Pregunta 4

Finalizado

Sin calificar

Dado el siguiente enunciado, especifique las user stories. Para cada una especifique tanto template como criterios de aceptación. Siempre que sea posible, refine las users stories en "sub" stories (indique explícitamente estas dependencias).

La facultad está desarrollando un sistema operativo propio y nos han solicitado que desarrollemos una aplicación que permita al usuario llevar su agenda de compromisos permitiendo múltiples calendarios (por ejemplo, "Cumpleaños", "Trabajo", "Partidos del mundial", etc.) creados externamente (por ejemplo, de Google o iCloud).

La principal funcionalidad de la agenda será permitir al usuario agregar eventos. Para agregar un evento se solicitará, obligatoriamente, informar el calendario, ya que la agenda tiene varios calendarios, en el cual se agregará el evento, el título del evento y el día y horario en el cual se realizará. Opcionalmente, el usuario podrá completar el lugar del evento y una lista de participantes (nombre y apellido). En caso que el nuevo evento se "solape" con otro, el sistema deberá informar esta situación, pero registrará el nuevo evento de todas formas. Además, al momento de registrar el evento el sistema deberá informar al calendario externo en el cual se agregó el evento para mantener consistencia entre la agenda y el calendario externo. Para ello, se comunica con el sistema de gestión de calendario.

El sistema deberá también permitir al usuario visualizar todos los eventos de sus calendarios. Esta funcionalidad permitirá filtrar los eventos por calendario, o rango de tiempo (día, semana o mes). Al visualizar un día en particular, el sistema deberá dar la opción al usuario de crear un nuevo evento.

Story: Agregar agendas

Descripcion: Como usuario quiero poder agregar mis agendas de otros servicios (Google) para así agregar mis eventos.

Criterio de aceptacion: El usuario debe poder agregar sus eventos de cualquiera de sus calendarios.

-Story: Ver calendario

Descripcion: Como usuario quiero ver mi agenda para así poder ver todos mis compromisos en un solo lugar

Substories:

- Agregar filtro para filtrar los eventos por calendario o rango de tiempo

- Al visualizar un día, crear evento

Criterio de aceptacion: Los usuarios deben poder ver su agenda y poder filtrar sus eventos

Story: Agregar eventos a calendario

Descripcion: Como usuario quiero poder agregar eventos para así tenerlo en mi agenda.

Criterios de aceptacion: El usuario debe poder agregar eventos a su calendario.

Pregunta 5

Finalizado

Sin calificar

¿Cuántos criterios de DONE debo definir para un proyecto?

Seleccione una:

- ☐ a. Uno por cada epica.
- ☐ b. Uno por cada Sprint
- ☐ c. Uno por cada tarea
- ☐ d. Uno para todo el proyecto.
- ☐ e. Uno por cada miembro del equipo de desarrollo
- ☒ f. Uno por cada user story.

Respuesta incorrecta.

La respuesta correcta es: Uno para todo el proyecto.

Pregunta 6

Finalizado

Sin calificar

Ordene los PBIs/requerimientos según el nivel de detalle. **Mayor** nivel de detalle (1) a **menor** nivel de detalle (4)

Tarea ▼

User Story ▼

Epica ▼

Feature ▼

Respuesta correcta

La respuesta correcta es: Tarea – 1, User Story – 2, Epica – 4, Feature – 3

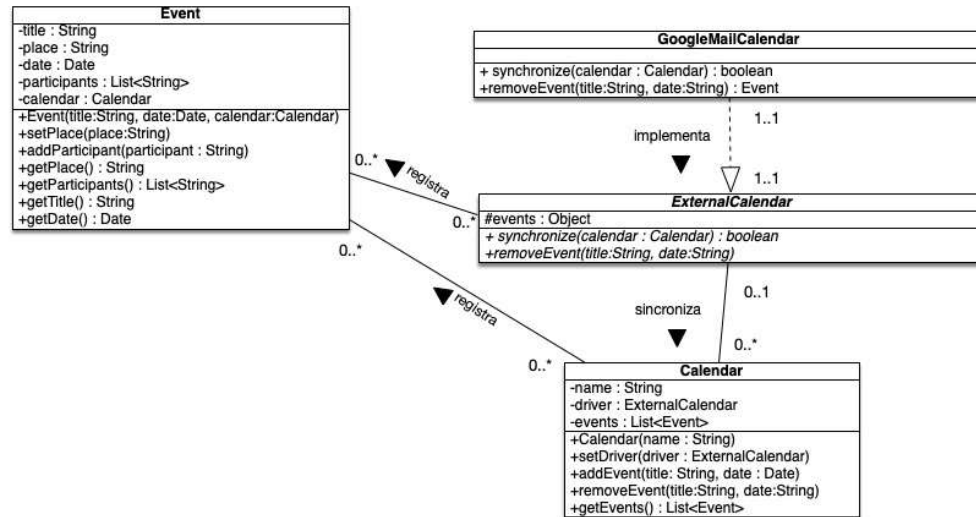
Pregunta 7

Finalizado

Sin calificar

A partir del siguiente diagrama, indique los errores. Solo tenga en cuenta el diagrama y la siguiente información:

- El calendario puede registrar varios eventos o ninguno
- Los eventos corresponden a un solo calendario



Event y ExternalCalendar no deberían tener relación

La variable protegida events, dentro de las variables del external Calendar no debería estar

La relación entre evento y calendario sería 1..1 y 0..n ya que los eventos solo corresponden a un calendario.

Errores mas importantes:

- Multiplicidad mal entre Event y Calendar.
- Falta dependencia entre Event y GoogleMailCalendar
- Relación entre ExternalCalendar y GoogleMailCalendar debe ser generalización
- No existe relación entre Event y ExternalCalendar
- La realización no lleva adornos

Pregunta 8

Finalizado

Sin calificar

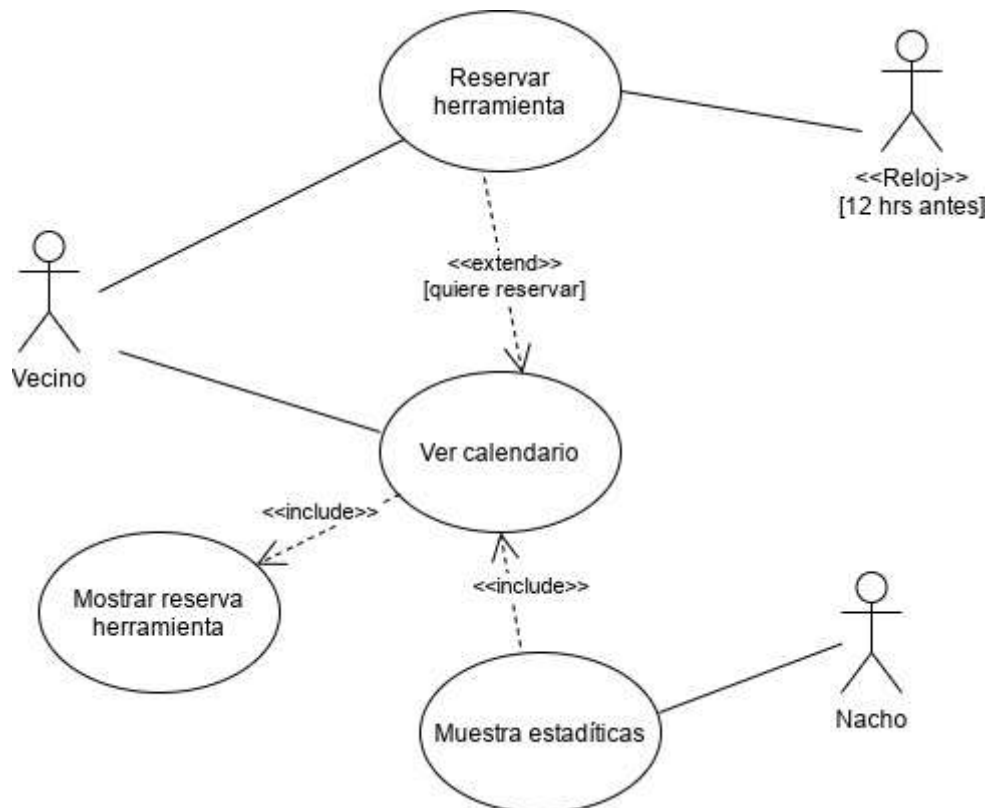
Dado el siguiente enunciado y el diagrama de Casos de Uso, indique los errores del diagrama de Casos de Uso.

El barrio Graduados de Tandil tiene un proyecto de herramientas compartidas (ej. Cortadora de pasto, escalera, hidrolavadora, etc.) entre los vecinos del barrio las cuales se guardan en un lugar común. Debido a la dificultad de saber dónde se encuentran las herramientas cuando un vecino no devuelve la misma a tiempo, Nacho, el responsable del proyecto, nos ha contactado para que los ayudemos a informatizar el proceso de reserva, obtención y devolución de las herramientas.

Nacho nos indica que necesitan una aplicación móvil que permita a un vecino reservar una herramienta. Para la reserva se le pedirá al vecino que seleccione la herramienta deseada de la lista de todas las herramientas existentes indicando el día y horario de comienzo y fin del préstamo. En caso de que la herramienta esté disponible en el tiempo solicitado, el vecino ingresará su nombre, apellido, email y dirección y el sistema reservará la herramienta deseada. Por una cuestión de organización, los préstamos pueden solicitarse como mínimo con 12hs de anticipación.

Con el objetivo de que todos los vecinos puedan saber qué herramientas están reservada y por cuánto tiempo, la aplicación además deberá mostrar un calendario indicando cada una de las reservas. Cuando el vecino accede a esta funcionalidad, la aplicación deberá brindarle la posibilidad de realizar una reserva en caso de que así lo desee.

Por último, con el objetivo de permitir a Nacho planificar mejor las próximas herramientas a comprar de acuerdo a las reservas solicitadas, la aplicación deberá presentar una página con las estadísticas de las reservas de cada una de las herramientas. Esta información será sólo accesible por Nacho.



El include en Mostrar reserva herramienta es incorrecto, bastaría solo con un activador al caso de uso. Ya que este solo se utiliza una vez.

Errores mas importantes a identificar:

- Nombre del actor Nacho no es un rol.
- Actor reloj no deberia estar dado que el chequeo de 12horas es parte de la funcionalidad del caso de uso
- No debe existir el caso de uso "Mostrar reserva herramienta"
- "Muestra estadisticas" no debe incluir a "Ver calendario"
- Mal el nombre del caso de uso "Muestra estadisticas". Deberia ser por ejemplo "Mostrar estadisticas"

Pregunta 9

Finalizado

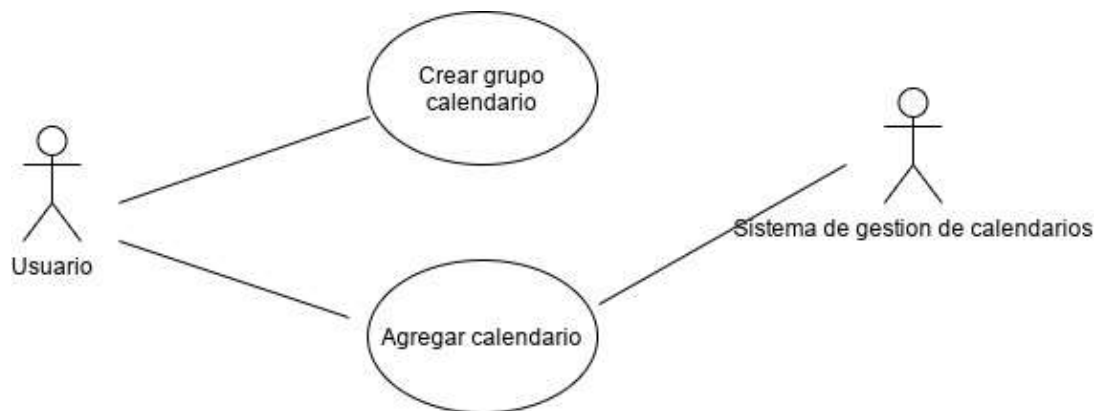
Sin calificar

Dado el siguiente enunciado y el diagrama de casos de uso incompleto, indique la relación existente entre los dos casos de uso.

La facultad está desarrollando un sistema operativo propio y nos han solicitado que desarrollemos una aplicación que permita al usuario llevar su agenda de compromisos permitiendo múltiples calendarios (por ejemplo, "Cumpleaños", "Trabajo", "Partidos del mundial", etc.) creados externamente (por ejemplo, de Google o iCloud).

El sistema permitirá agregar calendarios a la agenda. Para esto, solicitará el ingreso del nombre del calendario, una breve descripción y una dirección de email y contraseña que permita acceder a los eventos del calendario externo a incluir. En ese momento el sistema deberá usar el email y contraseña para conectarse con el sistema de gestión de calendarios para poder importar los eventos del calendario.

Para una mejor organización, la agenda permitirá crear grupos de calendarios. Para esto, el sistema listará todos los calendarios de la agenda, entonces el usuario podrá seleccionar los calendarios a agrupar y bajo qué nombre. En caso que el usuario quiera incorporar al grupo un calendario que no está en la agenda, el sistema informará que dicho calendario no existe.



Seleccione una:

- ☐ a. No hay relación
- ☒ b. "Agregar calendario" extiende a "Crear grupo calendario"
- ☐ c. "Crear grupo calendario" extiende a "Agregar calendario"
- ☐ d. "Crear grupo calendario" incluye a "Agregar calendario"
- ☐ e. "Agregar calendario" incluye a "Crear grupo calendario"

Respuesta incorrecta.

La respuesta correcta es: No hay relación

Pregunta 10

Finalizado

Sin calificar

Dado el siguiente diagrama de secuencia "incompleto", infiera cuáles son las clases involucradas y qué métodos contiene cada clase. Solo tenga en cuenta las clases y métodos indicados en el diagrama. Para ello utilice el siguiente template:

- Clase <nombreClase>
- Métodos clase <nombreClase>: <metodo1>, <metodo2>, ..., <metodoN>

En cada método indicar tipos de parámetros y retorno cuando corresponda.

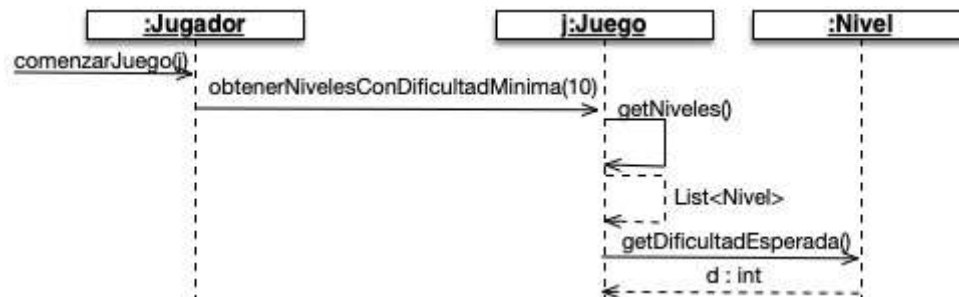
Ejemplo:

Clase A

Métodos clase A: int foo1(), foo2(int, int), foo3(String)

Clase B

Métodos clase B: bar1(Object), boolean bar2(Biblioteca)



- Clase Jugador
- Métodos clase Jugador: obtenerNivelesConDificultadMinima(int dificultad) : List<Nivel>

Clase Juego

Métodos clase Juego getNiveles(): List<Nivel>, getDificultadEsperada(): int

Métodos clase Jugador: comenzarJuego(Juego)

Métodos clase Juego: obtenerNivelesConDificultadMinima(int), List<Nivel> getNiveles()

Métodos clase Nivel: int getDificultadEsperada()