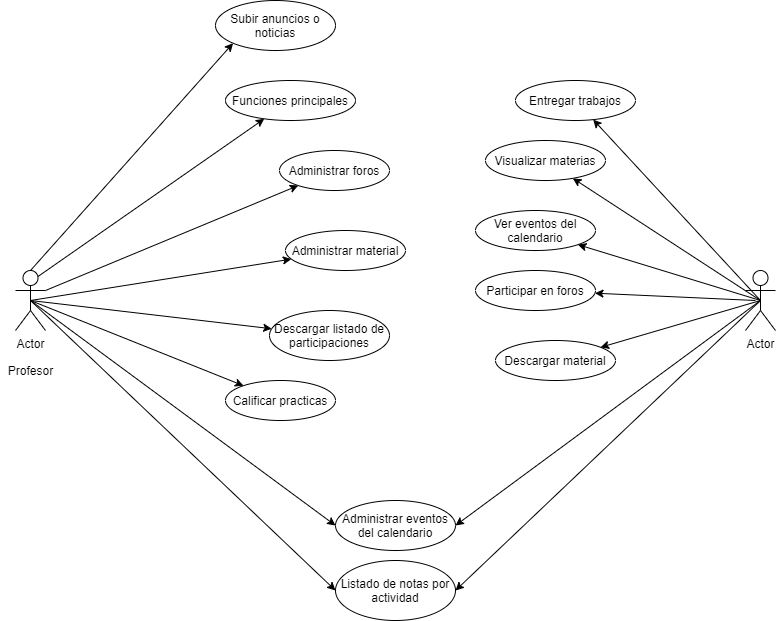
**INGENIERÍA DEL SOFTWARE**

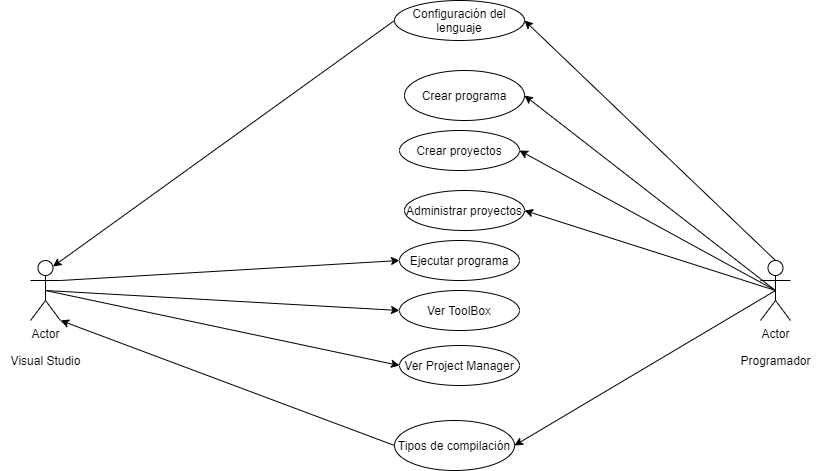
**CICLO 2020**

**TRABAJO PRÁCTICO 1 – CICLOS DE VIDA / CAPTURA DE REQUISITOS** *Ejercicios 1-9: HACER EL MODELO DE CASOS DE USO SENCILLO*

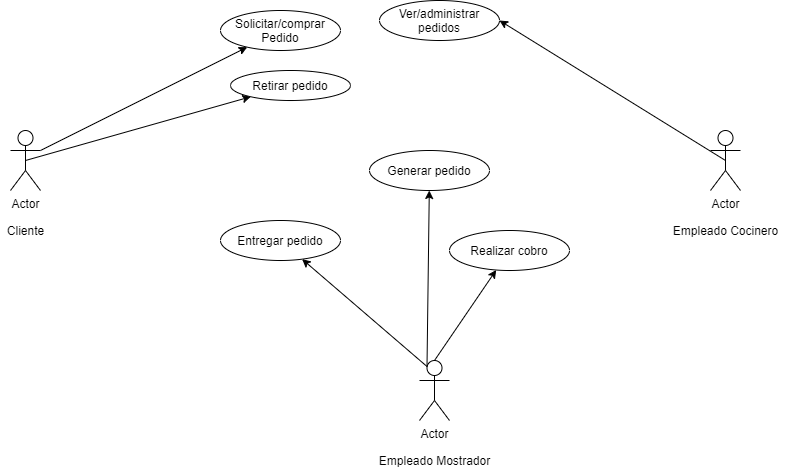
1. El nuestro Campus, los docentes podemos además de las funciones principales, crear foros, moderarlos, subir o anular material teórico-practico, descargar un listado por participación, y calificar prácticas. Los alumnos pueden entregar trabajos, visualizar materias, ver eventos de calendario, participar en foros, descargar material, entre otras. Se debe poder cargar eventos al calendario o darlos de baja y obtener listados de notas por actividad, subir anuncios o Noticias.



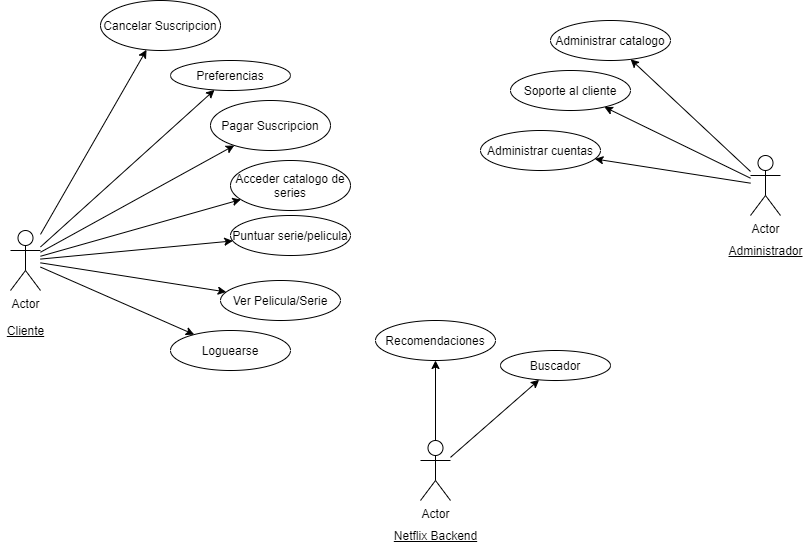
1. En visual studio, se pueden crear programas de varios lenguajes que deben ser configurados por el usuario. Se pueden crear proyectos, editarlos, abrirlos o cerrarlos. Además, se puede hacer la compilación, común o paso a paso, ejecutar programa, o ver paleta ToolBox y ProjectManager.



1. En Berguer King, les venden a los clientes los pedidos que podrán solicitar a través del mostrador. Luego los empleados entregan los pedidos según el numero asignado de espera. Las personas en la cocina son capaces de ver pedidos en pantalla. Pueden, además, marcarlos como preparados(amarillo), y una vez finalizados(verde) lo marcan en pantalla. Se les vende a los clientes por automac, donde los empleados les entregan los pedidos pedidos por ventanilla.



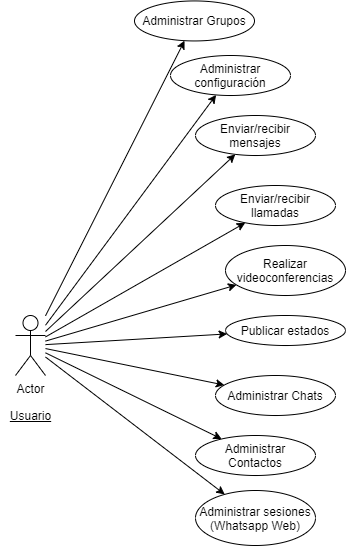
1. Investigue como se realiza una suscripción a Netflix, diagrame los actores y funciones básicas sobre la carga de contenido, reproducción y cobro mensual.



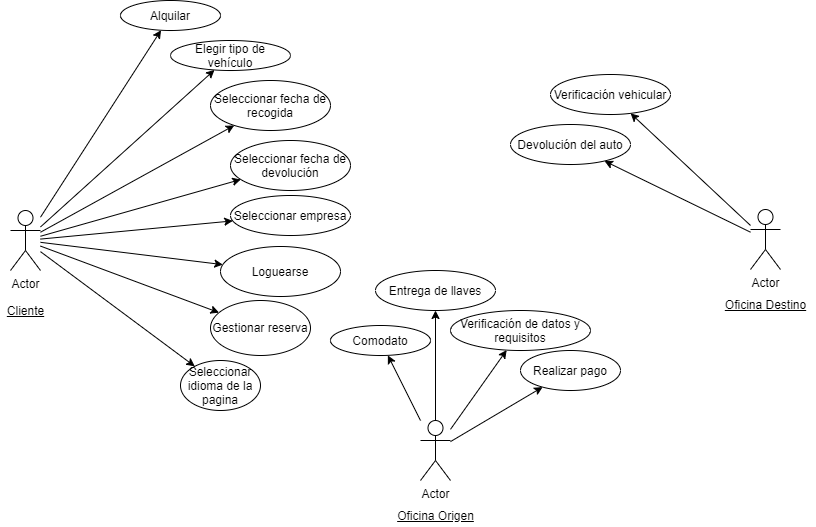
1. Investigue como se realiza una suscripción a Spotify, diagrame los actores y funciones básicas en cuanto las listas de reproducción y cobro mensual.



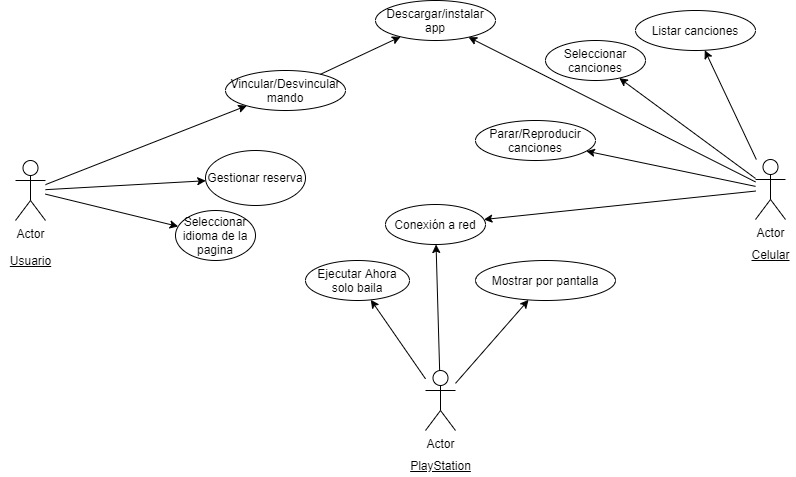
1. Realice el diagrama de casos de uso de la aplicación whatsapp en forma completa.



1. Una persona desea alquilar un auto a través de un Rental-Car que va desde Mendoza a Buenos Aires, investigue los procedimientos y pasos para su alquiler y devolución.



1. ¡Ahora Solo Baila! Es una aplicación de Play station que consiste en imitar canciones por pantalla. Para esto, se debe vincular un mando a distancia, usando el celular con la app del juego. El usuario debe vincular el mando, y luego permitirá listar canciones, seleccionar canciones, e incluso empezar una canción o pararla. Además, se debe poder desvincular el mando.



1. Amonzon! Es una plataforma donde los vendedores envían productos a sus compradores. Ellos cargan productos, los dan de baja, o los editan. Amonzon se encarga del envío de los paquetes al domicilio escogido. El comprador elige y compra productos, luego espera. Cada vendedor lo lleva hasta la sucursal de distribución. Se pueden devolver en 15 días gratis.



1. Se debe estimar el ***A)*** El tiempo total, y su costo. Se cuenta con 2 programadores y 2 analistas. El valor de hora (8 x día) programador es 6,5U$S y analista 7,5U$S. ***B)*** Costo total.

Tiempo total:

**En días: 196**

Captura (1AS) = 10 \* 7,5$US \* 8h = **600$US**

Analisis (2As) = 15 \* 15$US \* 8h = **1800$US**

Diseño (1AS) = 45 \* 7,5$US \* 8h = **2700$US**

Implementacion (2P) = 75 \* 13$US \* 8h = **7800$US**

Pruebas (1AS + 1 P) = 45 \* 14$US \* 8h = **5040$US**

Mantenimiento (1P) = 6 \* 6.5$ \* 8h = **312$US**

**Total = 18.252$US**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Captura | Análisis | Diseño | Implementación | Pruebas | Mantenimiento |
| 10 | 15 | 45 | 75 | 45 | 6 |
| Analista | Analistax2 | Analista | Programador x2 | Analista/Prog | Programador |

1. Se debe estimar el ***A)*** El tiempo total, y su costo. El valor de hora (8 x día) programador es 6,5U$S y analista 7,5U$S. ***B)*** Costo total. Para un ciclo INCREMENTAL, ***estime el valor de 3 los*** ***incrementos iguales + Captura única.*** Trabajaran 2 programadores Jr. y 1 analista en cadaincremento mensual.

Tiempo total:

**En días: 39**

Con incremental, 3 incrementos iguales de 25 dias.

Captura + esbozo (4) = 4 \* 7,5$US \* 8h = **240$US**

Cada Incremento (25) = (5\*15$\*8h) + (7\*7,5$\*8h) + (15\*13$\*8h) + (8 \* 14$\* 8h) =

= **600$US + 420$US + 1560$US + 896$US** =

= **3476 $US**

Todo el proyecto = 3 \* 3476$US = 10428$US + 240$US =

**Costo total: 10668$US**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Captura Única | Análisis | Diseño | Implementación | Pruebas |
| 4 | 5 | 7 | 15 | 8 |
| Analista | Analistax2 | Analista | Programador x2 | Analista/Prog |



**INGENIERÍA DEL SOFTWARE**

**CICLO 2020**

**TRABAJO PRÁCTICO 1 – CICLOS DE VIDA / CAPTURA DE REQUISITOS**

12. Completar con una (X). Categorías Alternativas: (verde, celeste y Naranja)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Lineal | Prototipos | Orientado  a objetos | Evolutivo | Code&Fix | Incremental | Cascada | Iterativo |
| No usa  prototipos  intermedios | **X** |  | X | **X** | **X** |  | **X** | **X** |
| Usa prototipos  Operativos |  | **X** |  |  |  |  |  | **X** |
| Prototipos no  funcionales |  | **X** |  |  |  |  |  |  |
| Necesita  requerimientos  completos | **X** |  | X |  |  | **X** |  |  |
| Parte de  requerimientos  incompletos |  | **X** |  | **X** | **X** |  |  | **X** |
| Se basa en una  serie de  pasadas |  |  |  | **X** |  | **X** | **X** | **X** |
| Versiones más  completas en  funcionalidades |  | **X** |  | **X** |  | **X** |  | **X** |
| Versiones  mejoradas de  todo el sistema |  | **X** |  | **X** |  | **X** |  | **X** |
| Versión única  terminada final  del proyecto | **X** |  | X |  | **X** | **X** | **X** |  |

13. UML

1. ¿Qué es y para que se usa UML?

**Visión general**

**“UML es un lenguaje de modelado de software éstandar”**

Suficientemente expresivo para cubrir distintas vistas de la arquitectura del software a lo largo del ciclo de vida.

Mayor nivel de abstracción que un lenguaje de programación, ya que es independiente del lenguaje de programación.

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

Nos indica cómo se pueden crear y leer modelos.

1. ¿Qué es un modelo? Y podría indicar ¿Para qué modelamos?

***“Un modelo es una abstracción de la realidad”***

Un modelo proporciona un plano de un sistema.

Incluye aquellos aspectos importantes del sistema y omite aquellos elementos menores que no son relevantes.

Cada modelo nos permite fijarnos en un aspecto distinto del sistema.

Un Modelo de Software es una Simplificación de la Realidad.

¿Para qué modelamos?

Construimos modelos para comprender mejor el sistema que estamos desarrollando.

Los Modelos:

Nos ayudan a visualizar cómo queremos que sea un sistema.

Nos ayudan a especificar la estructura o el comportamiento de un sistema.

Nos guían en la construcción de un sistema.

Documentan las decisiones que hemos adoptado.

Construimos modelos de sistemas complejos porque no podemos comprender el sistema en su totalidad.

Hay límites de la capacidad humana para comprender la complejidad.

A través del modelado, se reduce el problema que se está estudiando, centrándose en un solo aspecto a la vez. (divide y vencerás)

1. ¿Qué entiende por Abstracción? De ejemplos de clases existentes en vida real.

1-La abstracción es el proceso intelectual a través del cual separamos mentalmente las cualidades particulares de varios objetos para fijarnos únicamente en uno o diversas características comunes.

2-La abstracción es un concepto donde se separa o se aísla toda información que no resulta útil o relevante, y esto se hace a través de un proceso mental. Dicha clasificación ocurre definiendo ciertas características, funciones o atributos de una entidad.

Ejemplos

**Al nombrar un gato**: Si alguien dice la palabra gato, automáticamente la persona oyente sabe lo que es un gato y que este animal maúlla, caza ratones, entre otras cosas. Por lo que cada vez que alguien menciona la palabra gato, no es necesario definir todas sus características, ya que las personas de inmediato realizan un proceso de abstracción definiendo al animal.

**Extraer las características de un objeto como una motocicleta:** Primero se definen que similitudes tienen todas las motocicletas y luego se toman factores como: marca, modelo, tamaño, rendimiento, entre otros.

**Estaciones del año:** Por medio de la abstracción las personas logran diferenciar las características de cada estación del año, agrupando cada una con una serie de conceptos.

**Consultas médicas:** En una consulta médica, el doctor va desechando la información innecesaria que dice el paciente y acumula mentalmente solo aquella que le aporta datos de la enfermedad del paciente.

**Problemas cotidianos**: Al resolver un problema de la vida cotidiana, se separa o se desecha (se abstrae) todo lo que no resulte relevante para su solución, centrando el pensamiento en las cosas que son importantes y que sí ayuden a resolver el problema.

1. Investigue sobre UML y la Ingeniería directa e inversa

El lenguaje UML comenzó a gestarse en octubre de 1994, cuando Rumbaugh se unió a la compañía Rational fundada por Booch (dos reputados investiga- dores en el área de metodología del software). ... Todas estas colaboraciones condujeron a la definición de la primera versión de UML.

UML puede usarse en las diferentes etapas del ciclo de vida del desarrollo

Objetivos de UML

1-VISUALIZAR

2-ESPECIFICAR software

3-CONSTRUIR software

4-DOCUMENTAR

UML es un lenguaje para VISUALIZAR los elementos de un gran sistema software.

Esto facilita:

– la comunicación entre los desarrolladores

– la comprensión de las soluciones (notación gráfica)

– el mantenimiento de las soluciones conceptuales a lo largo del tiempo (ya que queda como documentación)

UML es un lenguaje para ESPECIFICAR software:

-Se pueden construir modelos precisos (no ambiguos) y completos.

– Cubre decisiones de análisis, diseño, etc.

UML permite especificar las características de un sistema antes de su construcción

UML un lenguaje para CONSTRUIR software

-No es un lenguaje de programación visual, pero sus modelos se pueden conectar de forma directa a una gran variedad de ellos.

– Correspondencias entre UML y lenguajes: Java, C++, etc.

UML es un lenguaje para DOCUMENTAR:

-requisitos, arquitectura, diseño, código fuente, pruebas, ...

Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Diagramas UML

-Diagrama de clases

- Diagrama de casos de uso

- Diagrama de secuencia

- Diagrama de colaboración

- Diagrama de estados

- Diagrama de actividades

- Diagrama de componentes

- Diagrama de despliegue

**Características que debe tener una herramienta UML**

**Ingeniería Directa:** Una herramienta UML no debe limitarse sólo a una representación pictórica de diagramas, sino que apoyar en forma directa y técnica la construcción de la aplicación en el lenguaje que se utiliza (Java, C++, ASP, ASPX, PHP). La ingeniería directa, va moviéndose desde los requerimientos, hacia el diseño (modelamiento, procesos) para llegar a la implementación. Nuestra experiencia, frente a la carencia de una herramienta UML es este aspecto, nos llevó a desarrollar RobotDocIRS, con el cual intentamos automatizar la generación de código fuente en forma robusta y pertinente a los intereses de cada proyecto.

**Ingeniería Inversa:** Es exactamente lo contrario de Ingeniería Directa. En la ingeniería inversa, la herramienta UML carga todos los archivos de la aplicación o del sistema, se identifican las dependencias entre las distintas clases, y, esencialmente, reconstruye la estructura de todo el requerimiento, junto con todas las relaciones entre las clases. Ingeniería Inversa es una característica normalmente proporcionada por sofisticadas herramientas UML. Es decir, se pretende utilizar el método como una aproximación práctica que permita generar modelos, utilizando el estándar UML, de aquellos sistemas cuya documentación es escasa, desactualizada o inexistente.

**Como funcionan ambas características en gráfico:**

