

## Calendario de Entregas

- 1) Primera Entrega: Fecha viernes 5 de junio
  - a. UML
  - b. Al menos 1 casos de uso
- 2) Segunda Entrega 1 semana: Fecha viernes 12 de junio
  - a. Estructura general y main del programa
    - i. En el archivo main.cpp solo puede estar el main. Todo lo demás debe estar en funciones.h y funciones.cpp
  - b. .h de todas las clases, y .h de la biblioteca de funciones creada.
- 3) Entrega Final: Fecha viernes 26 de junio
- 4) ¡Recuperatorio a Definir! Generalmente... previo a segunda fecha de final.

**SE DESCUENTA 1 PUNTO POR ENTREGA FUERA DE FECHA Y POR SEMANA DE ATRASO DE LA NOTA TOTAL POSIBLE A OBTENER. LA ENTREGA DE ALGO INCOMPLETO SE TOMA COMO QUE NO SE ENTREGÓ.**

### 1.1 El diseño del programa

<b>Main: Programa de simulación debe realizar una simulación del programa pedido. Debe:</b> <ul style="list-style-type: none"><li>● Instanciar todas las clases diseñadas (siempre que se pueda...).</li><li>● Utilizar todas las funciones / métodos que se hayan diseñado.</li><li>● Mostrar la herencia efectuada.</li><li>● Mostrar el polimorfismo diseñado.</li><li>● Mostrar Sobrecarga efectuada.</li><li>● Control de excepciones</li><li>● El main, puede tener una parte que requiera entrada del usuario.<ul style="list-style-type: none"><li>○ Debe tener una simulación autónoma.</li></ul></li></ul>	<b>Dentro de los atributos y parámetros obligatorios se debe contar con al menos</b> <ul style="list-style-type: none"><li>● 1 Atributo Constante</li><li>● 1 Atributo static</li><li>● 1 Atributo puntero doble y su buena utilización de memoria (new y delete)</li><li>● Control de excepciones</li><li>● Utilización del algoritmo random</li><li>● Al menos una función y una clase friend</li><li>● Sobrecarga de cin y cout para clase base y e hijas</li><li>● Sobrecarga de (al menos 1 vez... no es necesario repetir... aunque pueden hacerlo)<ul style="list-style-type: none"><li>○ &lt;, &gt;, ==, !=, ++ (pre y/o post), =, [], +, -.</li></ul></li><li>● Herencia</li><li>● Al menos 1 método polimórfico</li></ul>
--	---

**1 Torre de Control – Aeroclub** El aeroclub de san Fernando está actualizando los sistemas de su torre de control. Para ello quiere realizar una automatización y le pide desarrollar un programa de asignación automática de aterrizaje y despegue de aviones. Asimismo, se debe conocer todos los detalles del vuelo (si es de carga o pasajeros, y en su caso cuanta carga o cuántos pasajeros)

La torre tiene una lista de todos los aviones del aeroclub, cuando un avión quiere despegar, éste le informa a la torre de control, se le asigna una pista y comienza con las maniobras de despegue. De igual forma, cuando quiere aterrizar, le informa a la torre de control y nuevamente se le asigna una pista y comienza con las maniobras de aterrizaje. Se debe conocer en todo momento los aviones en vuelo, aterrizando, despegando y la cantidad de pistas libres.

Los aviones que tienen permitido volar hacia y desde el aeropuerto de San Fernando son los tipos **CESSNA** y **Biplanos de competición**. Cada tipo de avión tiene una forma de despegue particular. En el caso de los CESSNA debe aumentar su velocidad hasta 200 km/h, bajar los alerones, prender el control automático de estabilización de vuelo. Para los biplanos la velocidad de elevación es de 250 km/h, y la inclinación de despegue de 25°.

Dentro de las características comunes de los aviones se puede comentar que ambos tienen Hélice (1 para los **CESSNA** y dos para los biplanos), el tanque de combustible de cada tipo de avión depende del modelo particular, por lo cual deberá de tenerse esto en cuenta ya que por medidas de seguridad la torre debe saber el tiempo máximo de vuelo del avión (esto permite saber si pasado el tiempo el avión no volvió se debe reportar un incidente). Ambos aviones tienen capacidad para llevar pasajeros (cuantos depende del modelo). Los CESSNA además permiten llevar carga hasta 1000Kg (también depende del modelo en particular). Los biplanos no pueden llevar carga ya que están diseñados para realizar acrobacias. La aceleración de cada avión también depende del modelo en particular, pero por ejemplo los CESSNA A1 tienen una aceleración de 20 m/s<sup>2</sup> y los modelo B2 50 m/s<sup>2</sup>

Dependiendo del tamaño del avión y de la pista, no cualquier pista puede ser utilizada por cualquier tipo de avión. Además, hay que tener en cuenta el tiempo/distancia que le lleva al avión aterrizar y despegar. El largo de la pista tiene que estar en concordancia con esto.

**Modele. Los distintos tipos de aviones, la torre de control y las pistas. Además de la funcionalidad requerida. Genere un Simulador (main) de manera de verificar el correcto funcionamiento de la Watch Tower**

**2 Tienda de electrodomésticos** La cadena de electrodomésticos Musimundo quiere realizar una actualización de su sistema que se programó en los años 70 y no contempla ciertas opciones.

Los productos están diferenciados por tipo / marca/ y es necesario que el comprador, conozca sus dimensiones y peso y por supuesto precio. De manera de simplificar la programación del nuevo sistema. La Versión V0.1 solo utilizara productos del tipo microonda, televisor, y heladera.

El sistema debe permitirle al vendedor buscar dentro de la lista de artículos por código / marca / tipo y nombre y desplegar una lista de los productos encontrados que coincidan con la búsqueda.

Una vez finalizada la compra, el vendedor remite el producto al despachante.

El despachante, como es uno solo para toda la tienda, tiene una lista de productos a entregar, y a quién debe entregárselos. Mediante un Código de operación se le puede pedir al despachante que despache (valga la redundancia) el producto. Antes de entregárselo al cliente, el despachante debe asegurarse del correcto funcionamiento de los productos y para ello debe probarlos. Para las heladeras, las enchufa, espera 10 minutos, y verifica que la luz de la heladera se prenda al abrir la puerta, y se apague al cerrarse, verifica además que la temperatura sea la configurada en el climatizador interno. A los televisores los enchufa, enciende, y conecta el cable de video, y realiza una prueba de aumentar y disminuir el sonido, cambiar de canal y que se visualice la programación. Para los microondas, verifica la luz al abrir/cerrar la puerta, prueba los distintos modos de calentamiento, configura la potencia al máximo y enciende el microondas por 1 minuto para verificar que funcione correctamente. En general, la prueba del control de televisor falla 1/10 veces que se aprieta el botón. Las heladeras tienen una tasa de falla de 1/20 al verificar la temperatura y los microondas una tasa de falla de 1/15 en la prueba de 1 minuto.

Al final del día se debe conocer la lista de los equipos que se vendieron, y el monto total recaudado. Por último, si hay equipo stock menor al stock mínimo (mínima cantidad de artículos que se quiere tener) se debe realizar una lista con ellos, donde la cantidad es la necesaria para que el stock final supere el stock mínimo en 5 unidades y se debe saber el costo de esa lista. Si el costo total no supera los \$20.000 o si la cantidad total de ítems es menor a 15 la lista es rechazada (se borra), de lo contrario es Aprobarla (se agregan las cantidades a stock).

**Modele. Los distintos tipos de empleados, tipos de artículos y listas de artículos con las que se trabajan. Además de la funcionalidad requerida. Genere un Simulador (main) de manera de verificar el correcto funcionamiento de Compu Mundo Hiper Mega Red. El sistema simulador genera un cliente por minuto, este puede realizar una compra de un artículo al azar, o puede no comprar nada.**

**3 Sistema de seguimiento de equipos médicos** La fundación Favalaro quiere inventariar y realzar un control de equipos con mantenimiento preventivo y correctivo. Para minimizar el arranque del nuevo sistema, solo se cargarán equipos de electrocardiograma, respiradores y mesas de anestesia. El sistema debe permitir rastrear la ubicación de cada equipo en todo momento, además de permitir buscar el equipo por nombre / código / o tipo y presentar un listado de lo encontrado.

Además de las características básicas de cualquier equipo como dimensiones, y pesos, se desea almacenar un código único para cada equipo y una breve descripción y su estado (en uso, en espera, fuera de servicio, en mantenimiento), lugar actual, y lugar donde se guarda.

Cada equipo cuenta con un “calendario” donde indica las fechas en las que debe hacerse un mantenimiento preventivo del mismo. El mantenimiento preventivo consiste en realizar una serie de pruebas de rutina al equipo que permite verificar su correcto funcionamiento, y así evitar desperfectos si los tuviese. Evitando así los más costosos mantenimientos correctivos. Estos por su parte, suceden cuando el equipo falla, y debe de repararse. Para cada equipo, las acciones efectuadas durante el mantenimiento preventivo son distintas. Para los electrocardiógrafos, se los enchufa a un circuito simulador, y se prueba la correcta visualización de las distintas derivaciones y modos de traficación. Para los respiradores, se controla que el flujo de salida sea igual al configurado, y se verifican las alarmas de alta y baja presión, y de flujo 0 (taponamiento). Finalmente, para las mesas de anestesia se controlan los volúmenes de flujo de los gases frente a los fijados y las alarmas externas de baja o alta frecuencia cardiaca, y nivel de sueño.

Además, se debe imprimir una alerta, si al final del día el equipo no se encuentra en su lugar regular. Una vez al mes se elige un equipo de forma aleatoria, para ser verificado (). El sistema debe permitir al cliente saber en todo momento los equipos que no se encuentran en su lugar, y donde se encuentra cada equipo.

Al final del día, el sistema debe generar una lista con todos los mantenimientos preventivos hechos y su costo estimado. Además, una lista de todos los mantenimientos correctivos que se deban hacer o hayan quedado pendiente de días anteriores y su costo total. Si el costo total es menor a U\$D2.000 o la cantidad de equipos es menor que 5 se desestima. De no ser así, se realizan las reparaciones y se paga con la cuenta corriente que tiene la fundación.

**Modele. Los distintos tipos de equipos, listas de equipos con las que se trabajan. Además de la funcionalidad requerida. Genere un Simulador (main) de manera de verificar el correcto funcionamiento del Sistema Seguidor de Equipos.**

**4 Lista de pasajeros y personal de cabina de un avión** La International Air Transport Association (IATA), desea implementar una nueva regulación para garantizar la seguridad de los pasajeros y la verificación de información sobre vuelo, unificando el sistema de acceso a un avión y el registro de todos los sucesos que ocurren durante el vuelo. Para ello pide implantar una lista con los siguientes detalles,

La lista está conformada por pasajeros, pilotos, azafatas/os, un comisario de abordaje que está oculto (Marshall). Cada persona a bordo del avión posee determinados privilegios y tareas dentro del avión dependiendo de su tipo. Los pasajeros, que pueden ser de clase turista, ejecutiva o primera clase tienen la función de ser llevados por el avión, dentro de los privilegios básicos que poseen es pedir bebida o comida, ir al baño y volver a su asiento, dormir, y llamar a la azafata/o. Los de clase ejecutiva, pueden también trabajar en sus notebooks, y los de primera clase, pueden hacerse un masaje y tomar champagne.

Por su parte, las azafatas/os se encargan de entregar las bebidas y comidas a las personas del avión, atender los llamados de los clientes, y de los pilotos del avión (que no son los mismo). En el caso de que un pasajero se descompense, estas son las encargadas de atenderlo. El piloto, se encarga de pilotear el avión, y de hacer anuncios en el altavoz, puede pedirle a la azafata que le traiga una bebida sin alcohol, y algo de comer, o que ella realice algún anuncio por altavoz o a un pasajero en particular. De manera de simplificar, el copiloto es un tipo de piloto, que además de las funciones básicas, se comunica con la torre de control para realizar los pedidos de despegue y aterrizaje. Por ley, en cada avión debe haber un Marshall, cuya función es velar por la seguridad del avión, y si llegara a haber un arrebato de un pasajero, este lo reduce y lo encierra en la mini prisión que posee el avión.

Antes de subirse al avión. La compañía aérea registra en la lista quienes son los tripulantes del mismo (i.e. pilotos, azafatas y comisario) y una lista de códigos de pasaje habilitados para entrar al avión. En el caso del comisario, se le asigna un asiento no reservado por un pasajero en forma aleatoria. El código es único y contiene 13 dígitos alfanuméricos compuestos por dos caracteres que indican el sector (TU es turista, BS es ejecutiva, y PC es primera clase) 8 dígitos numéricos con el DNI del pasajero, y 3 dígitos más, 2 indicando fila (1-99) y por ultimo una letra (A-J) indicando el asiento de la fila.

Luego, a la entrada de la puerta, una lista de posibles “pasajeros” presentan su CODIGO de pasaje, si el código es válido, entra al avión (se añade a la lista), sino es rechazado (deja de existir para el sistema).

Al finalizar el vuelo, se debe imprimir la lista de eventos (pedidos de comida o bebida, etc) sucedidos en el vuelo.

**Modele. Los distintos tipos de pasajeros, tipos de códigos y listas con las que se trabajan. Además de la funcionalidad requerida. Genere un Simulador (main) de manera de verificar el correcto funcionamiento de MegaThunderPlaneList.**

**5 Línea de colectivos** La línea 60, una de las más grandes de la capital desea unificar el sistema de seguimiento de vehículos ya que ha habido muchos robos de colectivos y en los colectivos, y quieren identificar el lugar donde pasó. En estos momentos la línea 60 cuenta con 2 tipos de colectivos y un subtipo. Colectivos Con aire y dirección eléctrica. Cuyo subtipo es el colectivo acordeón. Y los colectivos viejos sin aire y dirección hidráulica.

Los colectivos tienen una determinada cantidad de pasajeros máxima, que está dada por su tipo. El colectivo, abre la puerta cuando llega a una parada, y deja primero bajar a los pasajeros que se deban bajar en esta parada y luego entrar a los pasajeros (debe verificar que la cantidad no supere a su capacidad, no se permite sobrepeso) que están en la parada. Cada pasajero debe indicar a la parada a la que viaja. El colectivo le cobrara dependiendo de una tabla que posee que va en relación con la cantidad de paradas entre la parada de subida y la de destino. Por obvios motivos, si el colectivo no pasa por la parada indicada por el pasajero, este no sube. Si en la cola de la parada se encuentra una persona con sillas de rueda, esta sube primero.

El colectivo posee además un recorrido, definido como una lista **estática** de paradas. Para poder identificar en todo momento al colectivo, el mismo tiene un sistema GPS que actualiza cada 5 minutos su posición.

Cuando un colectivo se rompe, se envía un colectivo que tenga poca gente arriba directamente hasta su ubicación, y se trasladan los pasajeros de uno a otro colectivo, y luego se lo lleva a arreglar al taller. Cuando el colectivo llega a su parada final, el sistema le asigna dinámicamente un nuevo ramal (obviamente, el ramal que se le asigna debe salir de la parada en la que está).

El sistema tiene además de la lista de colectivos, la lista de paradas. Estas además de su código identificador, tienen un nombre descriptivo, una dirección, y una lista de personas que están en la parada.

El sistema simulador tiene un método TICK que se llama una vez por minuto actualiza el estado de todos los objetos.

Al final del día, se debe desplegar una lista indicando la cantidad de personas que viajaron y el monto total colectado por cada colectivo (indicando ramal y código de colectivo), y finalmente el monto y la cantidad totales de pasajeros que viajaron en el día.

**Modele. Los distintos tipos de colectivos, listas de paradas con las que se trabajan y personas. Además de la funcionalidad requerida. Implemente un Simulador (main) de manera de verificar el correcto funcionamiento del Sistema Big Brother Bus, genere aleatoriamente personas en las paradas y simule el recorrido de los colectivos.**

**6 Cadena de cervecerías** Una cervecería ha decidido expandirse a lo largo de todo el país para ofrecer sus productos y precisa un sistema de seguimiento y control para los distintos locales que tendrá. Cada local, se identifica por un número, nombre y ubicación. La cadena tiene dos tipos de locales: PuntodeVenta, el cual solo tiene vendedores, y Bar que además de vendedores tiene meseros y mesas para atender. Todos los locales poseen un Encargado.

Los empleados, se identifican por un numero de CUIT, y poseen un salario por hora dependiendo de su tipo.

El encargado del local debe verificar que nunca va a quedarse con menos de M barriles de cada tipo de cerveza que posee la cadena. En el caso de que la cantidad de barriles sea menor a M, debe solicitarse a la central que reponga lo correspondiente. A su vez, en cada local se lleva una cuenta de los litros vendidos y el dinero ganado, así como debe llevarse la cuenta de las horas trabajadas de cada empleado para hacer el pago correspondiente a cada uno a fin de mes (algunos de los meseros no trabajan todos los días)

Para poder saber las horas exactas que trabajo cada uno, se registra la entrada y salida del mismo.

El sistema simulador tiene un método TICK que se llama una vez por “hora”. Este método llama a su vez al método SimularClientes() de cada local y que actualiza sus suministros y sus ganancias.

El método simularcliente de un punto de venta... simula que llegaron N clientes, pidieron X cerveza cada uno, y de van. Por su parte para Bar, la simulación es muy similar... pero además se ensucian mesas y los mozos deben limpiarla.!

El método virtual solicitarCerveza (tipo, cantidad) funciona dependiendo del tipo de local. Invente, como mínimo actualiza la cantidad de cerveza que tiene el local.

Al final del día, se debe desplegar una lista indicando la cantidad de litros vendidos y el monto total recolectado por cada local (indicando el nombre del local, su ubicación y su código), y finalmente el monto total.

**Modele: los distintos tipos de locales, la lista de locales que posee la cadena y la lista de empleados. Además de la funcionalidad requerida. Implemente un Simulador (main) de manera de verificar el correcto funcionamiento del sistema Cadena de cervecerías artesanales.**

**7 Covid outbreak** Debido a la conocida pandemia, el gobierno nacional nos ha dado la tarea de realizar una simulación para gestionar la distribución de personal asistencial a lo largo del país, en relación con la cantidad de casos que surgen diariamente en cada provincia o zona. En este primer piloto, nos ocuparemos de las zonas: CABA (VILLA URQUIZA y VILLA CRESPO), GBA NORTE (SAN ISIDRO y PILAR) y SUR (AVELLANEDA Y BANFIELD), SANTA FE y CÓRDOBA. Y debemos tener en cuenta a los siguientes actores:

- PERSONAS

- HABITANTES

- NORMALES

- RIESGO

- ASISTENCIAL

- MÉDICOS.

- ENFERMEROS.

La simulación deberá comenzar con una asignación aleatoria de Habitantes normales, de riesgo y personas ASISTENCIAL por zona. Se debe configurar el estado inicial de los HABITANTES:

- Probabilidad que estado HABITANTE NORMAL sea contagiado =

**P1%**.

- Probabilidad que estado HABITANTE RIESGO sea contagiado =

**P2%**.

Se desea saber el estado de cada habitante (contagiado, sano o fallecido) y en respuesta a este, se deberá desplegar y reorganizar las unidades asistenciales. Cada día pueden darse las siguientes situaciones:

- Cada HABITANTE contagiado, puede tener interacción con un número de entre **N<sub>min</sub>** y **N<sub>max</sub>** personas de la misma zona y la probabilidad de contagiar a esa persona se calculará de la siguiente forma:

- Si ambas personas son normales, hay un **P3%** de probabilidad.

- Si ambas personas son de riesgo, hay un **P4%** de probabilidad.

- Si un habitante es de un tipo y el otro habitante del otro, hay un **P5%** de probabilidad.

- Cada Persona contagiada, tiene una cierta posibilidad de fallecer. Esta posibilidad comienza siendo 0 en el primer día de contagio y aumenta un **P6%** cada día si es habitante normal y un **P7%** si es habitante de riesgo.

- Cada Persona Asistencial, tiene la posibilidad de “curar” a una persona contagiada que esté en la misma zona. El médico tiene un **P8%** de probabilidades de curarlo y el enfermero tiene un **P9%** de probabilidades de curarlo.



- Las unidades asistenciales pueden moverse entre zonas, pero tardarán 1 día. por lo cual, para las actividades diarias, no son tenidos en cuenta las nuevas unidades, ni las que se deben mover. Deben moverse según necesidades (establecer algún criterio).
- Todos los días, en el país se entrenan  $N_{asis}$  nuevas personas asistenciales. Y en cada zona “Nacen” de  $M_{min}$  a  $M_{max}$  personas nuevas de los diferentes tipos propuestos.

Se deberá presentar una simulación inicial, con al menos 400 objetos de tipo Habitante y al menos 50 personas asistenciales, distribuidos aleatoriamente.

Si se anima, establezca parámetros de modelización sacados de la realidad (ej. 1 médico cada 10 enfermeros), proporciónelos en relación con la cantidad de objetos a modelizar y/o realice un análisis sobre si es correcta o no la proporción, en comparación con las necesidades **Las propiedades, de las diferentes clases y subclases, y la forma de manejar, tanto la propagación como la contención del virus, quedan a cargo del alumno, su imaginación, o su búsqueda bibliográfica. En la primera entrega, deberá informar estas decisiones.**

**8 Labflix** Damian se volvió loco! Quiere hacerle la competencia a Netflix y por eso, les pide a ustedes (mano de obra barata) que hagan un servicio de streaming de VIDEO, AUDIO y ... para ser mejores que Netflix... JUEGOS.

La plataforma deberá permitir varios tipos de clientes (FREE, BASIC, PREMIUM) con acceso a diferentes tipos de contenidos y tiempo de uso (diario y continuo)

Una vez logueado, el usuario podrá listar una serie de servicios a los cuales pueda acceder., y luego seleccionarlo. Cada uno de estos servicios tendrá algunas acciones básicas comunes, y otras que solo pueden ellos. Ej.. los premium pueden “descargar” para mirar sin internet.

Ej. Todos los servicios se pueden iniciar, pausar y apagar. Los videos y las canciones además tienen el fast forward y backward, y record (solo basic y premium). Los juegos por su parte tienen para guardar el estado actual y otras actividades típicas asociadas a los juegos.

Para los usuarios FREE, todos los días hay una lista nueva de títulos a escoger.

Al final de cada semana, se debe poder visualizar un resumen de

- estadísticas, Mas vistos/escuchados/jugados por día y total semanal. Tiempo de reproducción promedio, etc
- Lo anterior, por tipo de usuario
- Para un usuario en particular

Los usuarios, pueden existir, pero estar “desconectados”. Se quiere saber

- Promedio de usuarios conectados por día
- Promedio de usuarios conectados en la semana

El contenido (juegos, pelis y música), deberá estar seccionado por país (y obvio un usuario tiene un país asignado)

## 9 Fabricación de cerveza

Usted se ha iniciado en el negocio de la fabricación de cervezas artesanales, y para llevar un buen control sobre sus producciones y ventas ha decidido realizar un programa que lo ayude. El mismo se encargará de tomar los datos de su equipo y simular las fabricaciones que se pidan en las distintas ordenes de pedido que le lleguen. La clase receta tendrá un nombre, IBU (índice de amargor por unidad), graduación alcohólica, ollas, fermentador e insumos. Como métodos tendrá Simular() que simulará la fabricación, guardando todos los datos en la misma e ImprimirInfo() que desplegara toda la información de la receta.

Las ollas a utilizar son 3. Todas poseen un sensor de temperatura, un material del cual están fabricadas (aluminio, acero inoxidable) y una capacidad. La primera olla es solo de agua. La segunda es de maceración, es decir, donde se vierte la malta por lo que posee un remo y un filtro (puede ser una bolsa de filtrado, falsofondo o chupapalmer). La tercera es de cocción, ahí se vierten el lúpulo y otros insumos, precisa de una espumadera para remover la espuma en la cocción.

Las ollas tienen el método polimórfico HacerAlgo() que obviamente depende del tipo de olla.

De todos los insumos se conoce su cantidad, costo y nombre. De estos se pueden derivar los lúpulos que además de esa información poseen Alfa-ácidos (importante para calcular los IBUs) y la malta que es quien aporta el color de la cerveza. Pueden adicionarse extras como miel para una Honey beer o clarificantes como el Irish mosh.

Por último se precisa tener modelizado un fermentador. El mismo posee una capacidad, tipo de enfriamiento, temperatura y volumen actual.

Para realizar la simulación se precisa definir la densidad luego de la cocción (Original Gravity) y la densidad al momento de embotellar, con esto se calcula la graduación alcohólica con la siguiente funcion:

**GraduacionAlcoholica = (OG – DensidadFinal)\*0,13125**

Por otro lado, es importante también conocer el IBU final de la cerveza. Para poder calcularlo se precisa conocer en que momento de la cocción fueron introducidos los lúpulos, en que cantidad y que alfa-acidos poseían. La función para calcular esto es:

**IBU = (Gramos x FA x %AA x 1000) / (Litros)**

Donde FA es el factor de aprovechamiento de alfa-acidos (puede verse en la tabla el factor para cada tipo de lúpulo) y %AA es el porcentaje de alfa-acidos que posee el lúpulo vertido. Litros se refiere a los que quedaron al final de la coccion, es decir, los que van al fermentador.

Finalmente deben sumarse los IBUs aportados por cada lúpulo para obtener el IBU final.

Tabla de FA:<https://cervezomicon.files.wordpress.com/2016/08/tabla-valores-de-aprovechamiento-todos.jpg?w=584>

Un pedido puede recibir distintos tipos de cervezas como así también el porcentaje alcohólico final de cada una y el picor de la misma, y una sublista de materiales adicionales a utilizar (miel, tabaco, etc)

**Modele: los distintos insumos, la lista de los mismos y los resultados de la receta. Además de la funcionalidad requerida. Implemente un Simulador (main) de manera de verificar el correcto funcionamiento del sistema Brewmaster.**

## 10 Game master

¡Se nos ha encargado la realización de un juego muy similar a T.E.G. donde dos o más jugadores deben apoderarse del MUNDO!

La diferencia principal respecto al TEG es que en este juego hay distintos tipos de unidades y un sistema de preferencia de ataque y defensa. Los CABALLEROS, tienen un CONTRATAQUE con un 25% de aumento de daño cada vez que se los ataca. Los caballeros son FUERTES frente a los ARQUEROS. Los arqueros son unidades que ALEATORIAMENTE pueden atacar una segunda vez con un 50% de aumento de daño y son fuertes contra los MAGOS. Los MAGOS son unidades que atacan en ZONA, es decir atacan a todas las unidades enemigas en el campo de batalla, pero sus puntos de daños son mucho más bajos que los de las demás clases. Por su parte los magos son fuertes frente a los CABALLEROS. Ser fuerte frente a otra clase significa que se obtiene un 25% de aumento de incremento en el ataque. Por su parte si se ataca en forma inversa, EJ un caballero ataca a un mago, su ataque disminuye un 25%

Los caballeros tienen un ataque (AT) entre 30-40 y entre 80 y 100 de vida (HP).

Los arqueros tienen un ataque (AT) entre 20-30 y entre 60 y 80 de vida (HP).

Los caballeros tienen un ataque (AT) entre 10-20 y entre 40 y 60 de vida (HP).

Cada jugador comienza con 5 países y 10 Tropa de unidades. Cada tropa es de un único tipo de unidad y de 10-20 unidades si son caballeros, 15-30 si son arqueros y 20-40 si son magos.

En cada turno, cada jugador puede realizar **hasta** 3 ataques, desde un país a un país vecino que no controle y atacar con la cantidad de grupos de unidades que desee, que estén, obviamente, en ese país.

En el ataque, Primero atacan todos los grupos del atacante, y luego, todos los grupos del enemigo que se encuentren en ese país, así hasta que alguno de los dos jugadores se quede sin unidades. **(Opción 1): Se ataca eligiendo aleatoriamente en cada ronda atacante y atacado, (Opción 2) En cada ronda de ataque cada jugador puede seleccionar tropa atacante y atacada.**

***Cuando un grupo ataca o defiende lo hace con la fuerza combinada de todos sus integrantes. Ej. Si tenemos un grupo de 3 guerreros cada uno con 4 5 y 6 de fuerza. el grupo ataca con 15 puntos de daño. Cuando un grupo recibe XX de daño lo reparte desde las unidades con menor HP hasta la de mayor. Cuando una unidad posee menos de 10% de 5 de vida se muere.***

Si el atacante gana (el atacado se queda sin tropas primero), el jugador gana el país, y puede mover la cantidad de tropas que desee a ese país.

Antes de comenzar la siguiente ronda, cada jugador obtiene **(Cantidad de países/2)** tropas aleatorias con las

mismas cantidades descritas anteriormente. El jugador debe indicar en qué país quiere colocar cada tropa y si quiere mantenerla como un grupo separado o sumarla a una tropa ya existente en ese país.

**Genere una partida de ejemplo con solo los países de Latinoamérica y divídalos de igual manera entre los dos jugadores.**

**Permita jugar hasta que uno de los dos conquiste el mundo o 10 turnos, lo que suceda primero (Permita quitar este limitante fácilmente por si quiere jugar hasta el final)**

**AYUDA: En la clase PAIS defina un atributo estático Listadepaises... donde estén todos los países creados.**