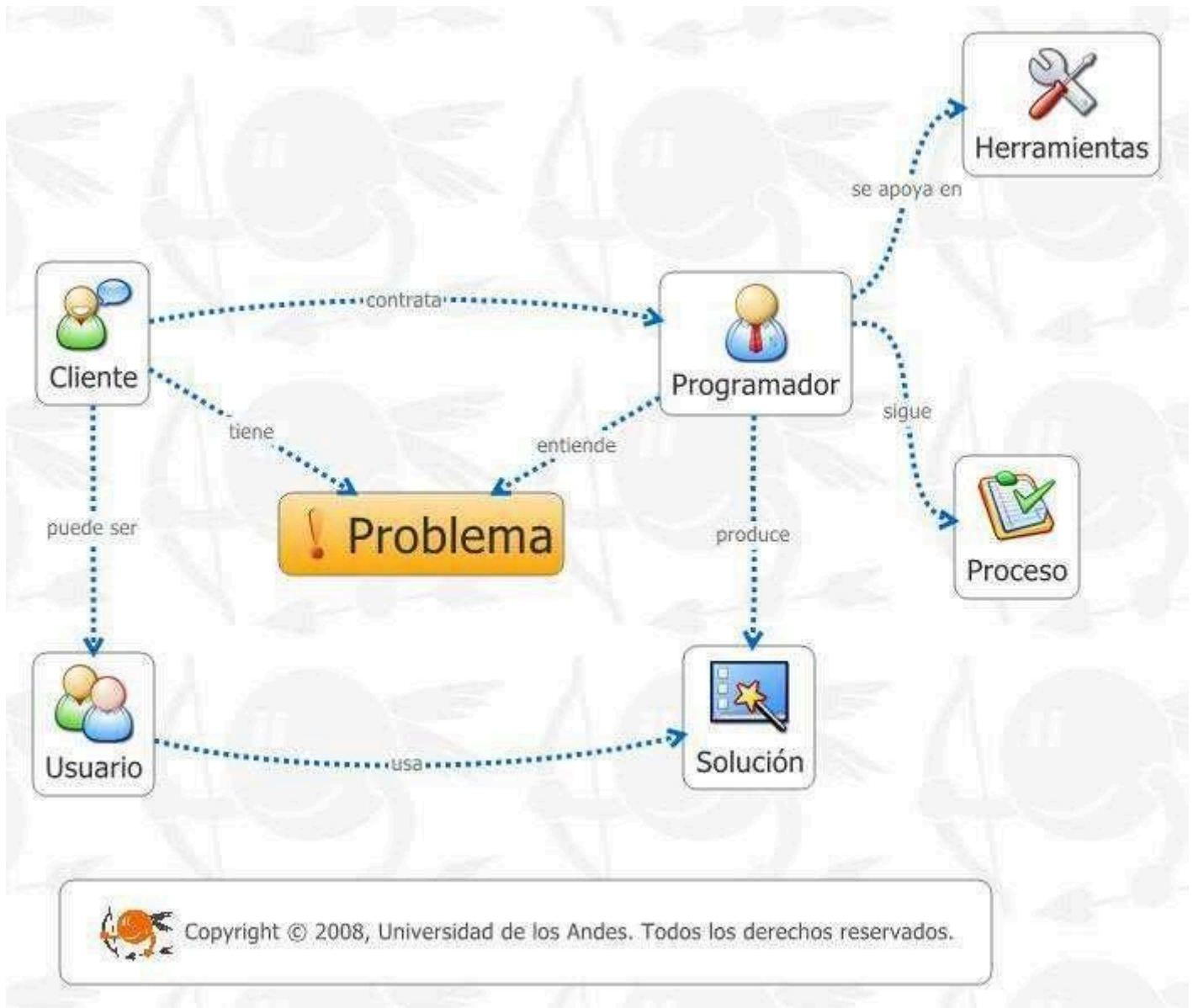


PROYECTO DE LA ASIGNATURA



Contenido

| | |
|---|--------------------------------------|
| ACTIVIDADES PRELIMINARES..... | 4 |
| CONTEXTO..... | 4 |
| EXPERIENCIA..... | 4 |
| ACCIÓN..... | 5 |
| EVALUACIÓN..... | ¡Error! Marcador no definido. |
| 1. WORKFLOW DE REQUERIMIENTOS..... | 8 |
| 2. WORKFLOW DE ANÁLISIS Y DISEÑO..... | 11 |
| 3. WORKFLOW DE IMPLEMENTACIÓN..... | 14 |
| 4. WORKFLOW DE PRUEBAS (TESTING)..... | 17 |
| 5. DOCUMENTACIÓN..... | 19 |
| BIBLIOGRAFÍA..... | 21 |

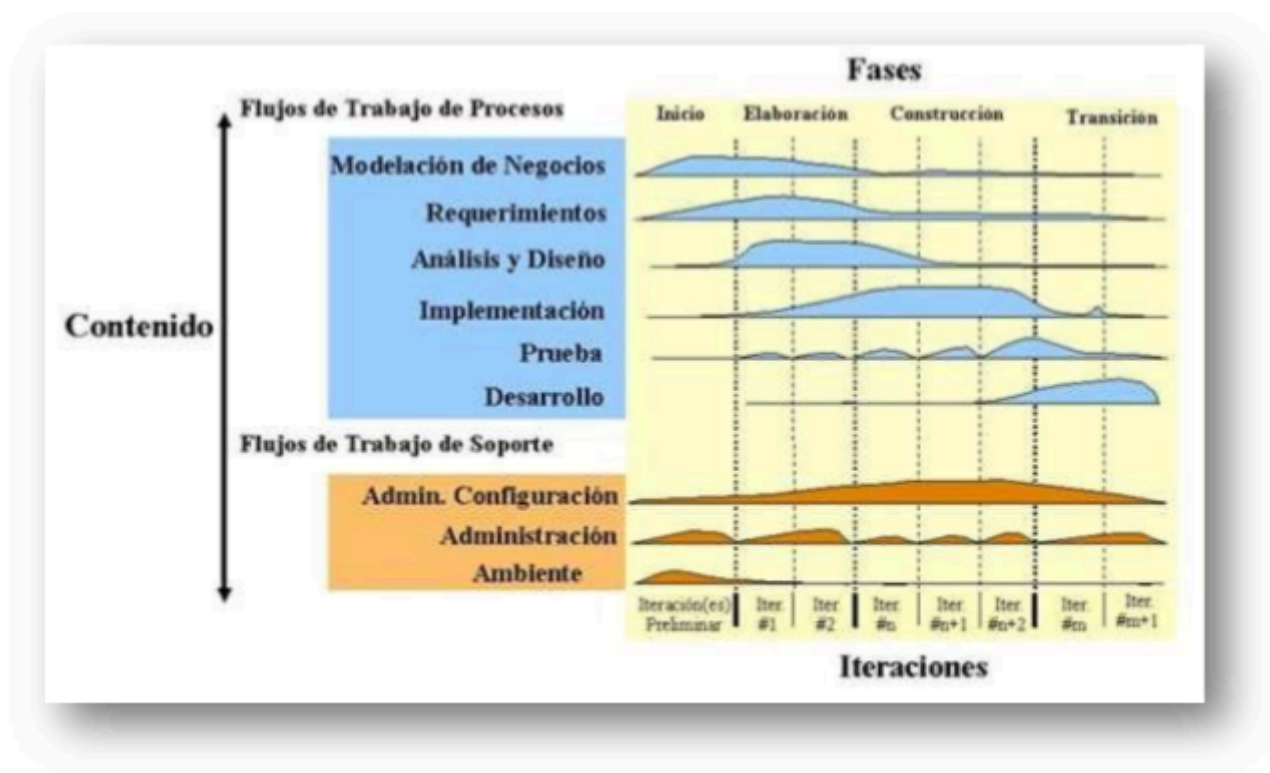
BITÁCORA

| Entrega | Fecha | Objetivo | Observaciones | Responsables |
|-------------------------------|-------------|---|---|--------------|
| WORKFLOW DE REQUERIMIENTOS | | | | |
| 1 | Entrega # 1 | Ejercicios hasta el literal "p" (pág. 10) (escanear respuestas) | "Sin portada no hay nota" "La elaboración se efectúa a mano". | |
| WORKFLOW DE ANALISIS Y DISEÑO | | | | |
| 2 | Entrega # 2 | Hasta el numeral 2.4. | "Sin portada con los responsables no hay nota en cualquier entrega" | |
| WORKFLOW DE IMPLEMENTACIÓN | | | | |
| 3 | Entrega # 3 | 3.2. | Versión 1 del código fuente | |
| 4 | Entrega # 4 | 4.2. | //Adjuntar comentarios con los cambios efectuados en el código fuente | |
| WORKFLOW DE PRUEBAS | | | | |
| 5 | Opcional | 5.2. | Documentos entregables del proyecto | |

METODOLOGÍA: PROCESO UNIFICADO (RATIONAL DE IBM) - R.U.P.

<http://creately.com/Draw-UML-and-Class-Diagrams-Online>

<https://www.gliffy.com/examples/uml-diagrams>



ACTIVIDADES PRELIMINARES

CONTEXTO

LA ESPECIFICACIÓN DE UN PROBLEMA:

- a. Resuma el ciclo de vida de construcción de un programa.
- b. Explique los aspectos que hacen parte del análisis de un problema.
- c. Explique las etapas del proceso de solución de problemas.
- d. ¿Cuáles son los elementos que se deben entregar a un cliente?

EXPERIENCIA


- e. Elabore la Tarea No. 1 (pág. 5 del texto guía), con el objetivo de identificar los aspectos que forma parte de un problema.
- f. Elabore la Tarea No. 2 (pág. 13), con el objetivo de identificar los requerimientos funcionales de un problema.
- g. Elabore la Tarea No. 3 (pág. 14), con el objetivo de identificar los requerimientos funcionales de un problema.
- h. Elabore la Tarea No. 4 (pág. 17), con el objetivo de identificar las entidades del mundo del problema.
- i. Elabore la Tarea No. 5 (pág. 20), con el objetivo de identificar las entidades de un caso de estudio.
- j. Elabore la Tarea No. 6 (pág. 23), con el objetivo de reflexionar sobre el nivel de precisión de un algoritmo.

RUBBY CASALLAS GUTIERREZ, JORGE ALBERTO VILLALOBOS SALCEDO, "Fundamentos de Programación: Aprendizaje Activo Basado en Casos" En: México 2006. Ed:Pearson Education ISBN: 970-26-0846-5

TRABAJO PRELIMINAR: CASO DE ESTUDIO

Para los siguientes ejercicios:

- Dirígete al portal: <https://cupi2.virtual.uniandes.edu.co/nivel-1/ejemplos-n1>
- Selecciona uno de los ejemplos propuestos:
 - El empleado, El Triángulo (r)
 - Simulador Bancario, Elecciones (par)



Ejemplo 3: La encuesta

✍ Curso: APO1

📖 Nivel: N1

📖 Descripción: Se quiere construir un programa para manejar los resultados de una encuesta de opinión.

🔗 Link: [Ir al ejemplo](#)

- Ingresa a la opción '[Ir al ejemplo](#)' del caso que decidan estudiar:
- k. Estudia los siguientes aspectos del ejemplo seleccionado: *Enunciado*, *Requerimientos funcionales (casos de uso)* y el *Modelo (clases del proyecto)*. A continuación, redacta el **enunciado del problema** y el **nombre cada uno de los requerimientos funcionales del proyecto**.
 - l. Dibuja el respectivo **diagrama de casos de uso** del ejemplo elegido (consulta “requerimientos funcionales” en ‘Archivo’).
 - m. Observa nuevamente el modelo conceptual del caso y escribe **el nombre de cada una las clases** identificando sus respectivas variables (atributos) y funciones:

Simulador bancario

Enunciado
Archivos
Modelo
Visualización de código

Mundo

CuentaBancaria

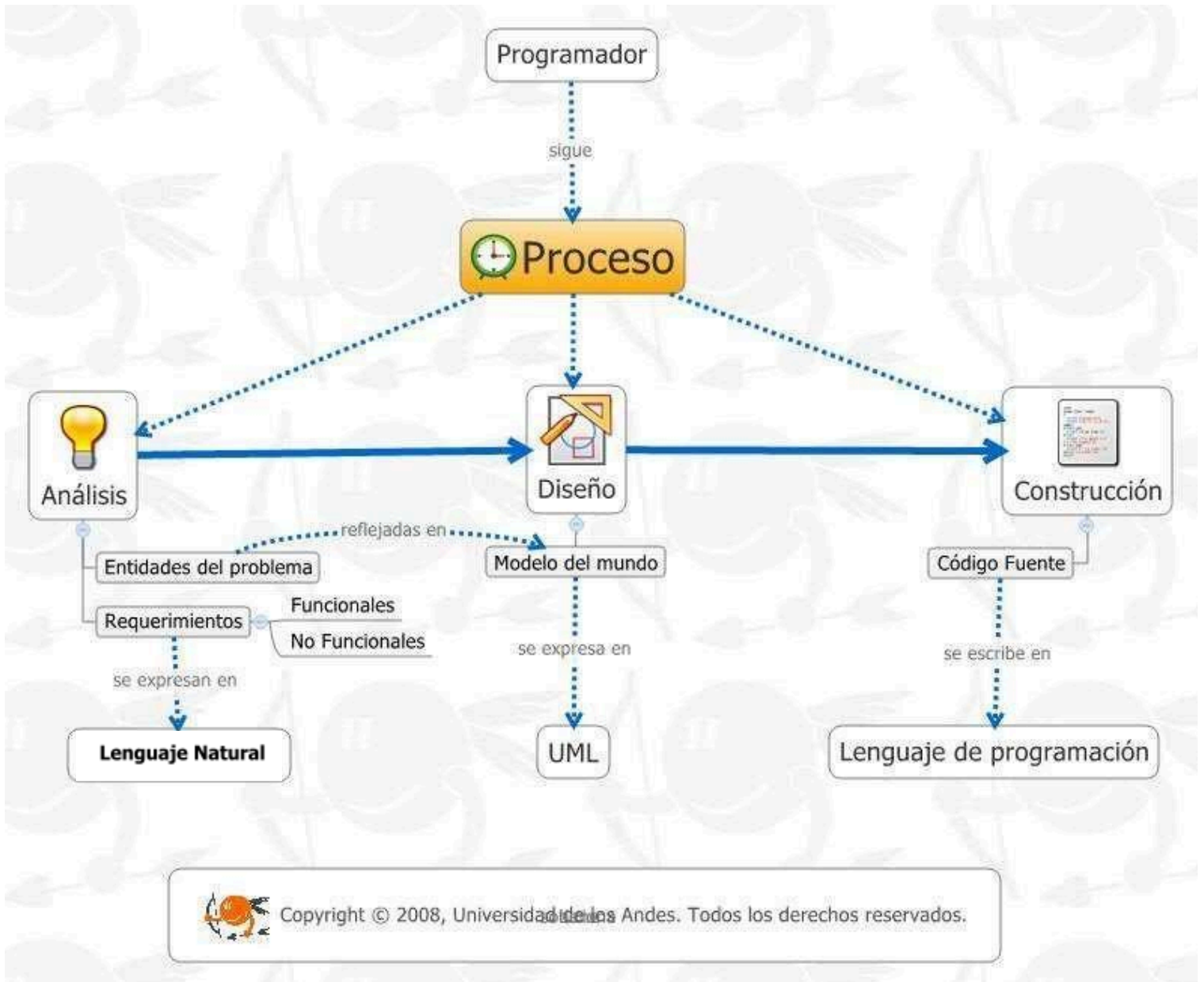
n. Debes plantear 2 ideas de proyecto (problemas solubles y algorítmicos): Idea 1:

| | | |
|---------------------------|-----------|--|
| Requerimiento funcional 1 | Nombre | |
| | Resumen | |
| | Entradas | |
| | Resultado | |
| Requerimiento funcional 2 | Nombre | |
| | Resumen | |
| | Entradas | |
| | Resultado | |
| Requerimiento funcional 3 | Nombre | |
| | Resumen | |
| | Entradas | |
| | Resultado | |
| Requerimiento funcional 4 | Nombre | |
| | Resumen | |
| | Entradas | |
| | Resultado | |

Idea 2:

| | | |
|---------------------------|-----------|--|
| Requerimiento funcional 1 | Nombre | |
| | Resumen | |
| | Entradas | |
| | Resultado | |
| Requerimiento funcional 2 | Nombre | |
| | Resumen | |
| | Entradas | |
| | Resultado | |
| Requerimiento funcional 3 | Nombre | |
| | Resumen | |
| | Entradas | |
| | Resultado | |
| Requerimiento funcional 4 | Nombre | |
| | Resumen | |
| | Entradas | |
| | Resultado | |

1. WORKFLOW DE REQUERIMIENTOS



1.1. Describa y justifique el problema, indicando porque se trata de un problema soluble (referencia pág. 5):

| | |
|----------------------------|--|
| Problema: | Se desea realizar una aplicación para el manejo de una mini empresa de transporte. La empresa tiene capacidad para administrar sus camiones y sus respectivas cargas .Se requiere registrar, consultar y gestionar la información de cada camión, asegurando que la carga asignada no supere su capacidad. Adicionalmente, se desea que la aplicación sea capaz de determinar el mejor camión disponible para transportar una carga determinada. Se considera el mejor camión aquel que tiene, entre los camiones que tienen una capacidad suficiente para transportar el peso dado, el menor consumo. |
| Cliente | Mini empresa de transporte |
| Usuario | Gestionar los camiones dentro de la empresa |
| Requerimiento funcional | 1)Registrar camiones con matrícula, capacidad de carga, consumo de gasolina y carga actual. 2) Consultar la información de un camión a partir de su matrícula 3) Cargar un camión verificando que no supere su capacidad máxima 4)Descargar un camión reduciendo su carga actual 5)Determinar el mejor camion disponible con menor consumo de gasolina para carga específica 6)Informar cuando no haya camión disponible para una carga dada. |
| Mundo del problema | Es necesario conocer la matrícula, capacidad de carga, consumo de gasolina y carga actual de cada camión. Se desea evaluar los camiones disponibles para seleccionar el más eficiente en términos de consumo de gasolina y capacidad de carga. |
| Requerimiento no funcional | La aplicación debe ser rápida y eficiente en la consulta y asignación de camiones. Debe ser fácil de usar para los encargados y garantizar precisión en los cálculos de carga y consumo de gasolina. |

1.2. Señale mínimo 4 requerimientos funcionales indicando el tipo de dato que existen en el lenguaje de implementación para las variables que serán utilizadas en las entradas y salidas del requerimiento (Pág. 11-12):

| | | |
|---------------------------|----------|--|
| REQUERIMIENTO FUNCIONAL 1 | Nombre | Registro de camiones |
| | Resumen | Permite registrar la matrícula, capacidad de carga, consumo de gasolina y carga actual de cada camión. |
| | Entradas | Matrícula(string), capacidad de carga(float), consumo de gasolina(float), carga actual(float). |
| | Salidas | Confirmación de registro |
| REQUERIMIENTO FUNCIONAL 2 | Nombre | Carga de camión |
| | Resumen | Verifica y permite agregar carga a un camión sin superar su capacidad máxima. |
| | Entradas | Matricula(string), carga a agregar(float) |
| | Salidas | Mensaje de éxito o error según la validación. |
| REQUERIMIENTO FUNCIONAL 3 | Nombre | Consulta de datos de camión |
| | Resumen | Permite consultar los datos de un camión a partir de su matrícula. |
| | Entradas | Matricula(string) |
| | Salidas | Información de camión, matrícula, capacidad de carga, consumo de gasolina y carga actual. |
| REQUERIMIENTO FUNCIONAL 4 | Nombre | Selección del mejor camión para carga |
| | Resumen | Identifica el camión con menor consumo de gasolina y capacidad suficiente para transportar una carga dada. |
| | Entradas | Carga requería(float) |
| | Salidas | Matrícula del camión seleccionado o mensaje de "No disponible" |

2. WORKFLOW DE ANÁLISIS Y DISEÑO

ANALISIS DE ATRIBUTOS:

2.1. Identifique las entidades del mundo problema, mínimo 3 (consulta como referencia la pág. 16, J. Villalobos)
RUBBY CASALLAS GUTIERREZ, JORGE ALBERTO VILLALOBOS SALCEDO, "Fundamentos de Programación: Aprendizaje Activo Basado en Casos" En: México 2006. Ed:Pearson Education ISBN: 970-26-0846-5

| ENTIDADES (Las que necesites) | NOMBRE | DESCRIPCIÓN |
|----------------------------------|--------|-------------|
| 1 | | |
| 2 | | |
| 3 | | |

2.2. Señale las características de las entidades descritas:

| NOMBRE DE LA ENTIDAD #1 | VALORES POSIBLES (Rango aceptable) | TIPO DE DATO (C++) Y EXPLICACIÓN |
|----------------------------|---------------------------------------|----------------------------------|
| | | |
| | | |
| | | |

| NOMBRE DE LA ENTIDAD #2 | VALORES POSIBLE S (Rango aceptable) | TIPO DE DATO (C++) Y EXPLICACIÓN |
|----------------------------|---|----------------------------------|
| | | |
| | | |
| | | |

| NOMBRE DE LA ENTIDAD #3 | VALORES POSIBLE S (Rango aceptable) | TIPO DE DATO (C++) Y EXPLICACIÓN |
|----------------------------|---|----------------------------------|
| | | |
| | | |
| | | |

| NOMBRE DE LA ENTIDAD #x | VALORES POSIBLE S (Rango aceptable) | TIPO DE DATO (C++) Y EXPLICACIÓN |
|----------------------------|---|----------------------------------|
| | | |

| | | |
|--|--|--|
| | | |
| | | |

2.3. Establezca las relaciones entre las entidades de forma lógica, en un esquema gráfico, estableciendo las entidades y las relaciones (investiga: Diagrama E/R) (Pág. 21-22):

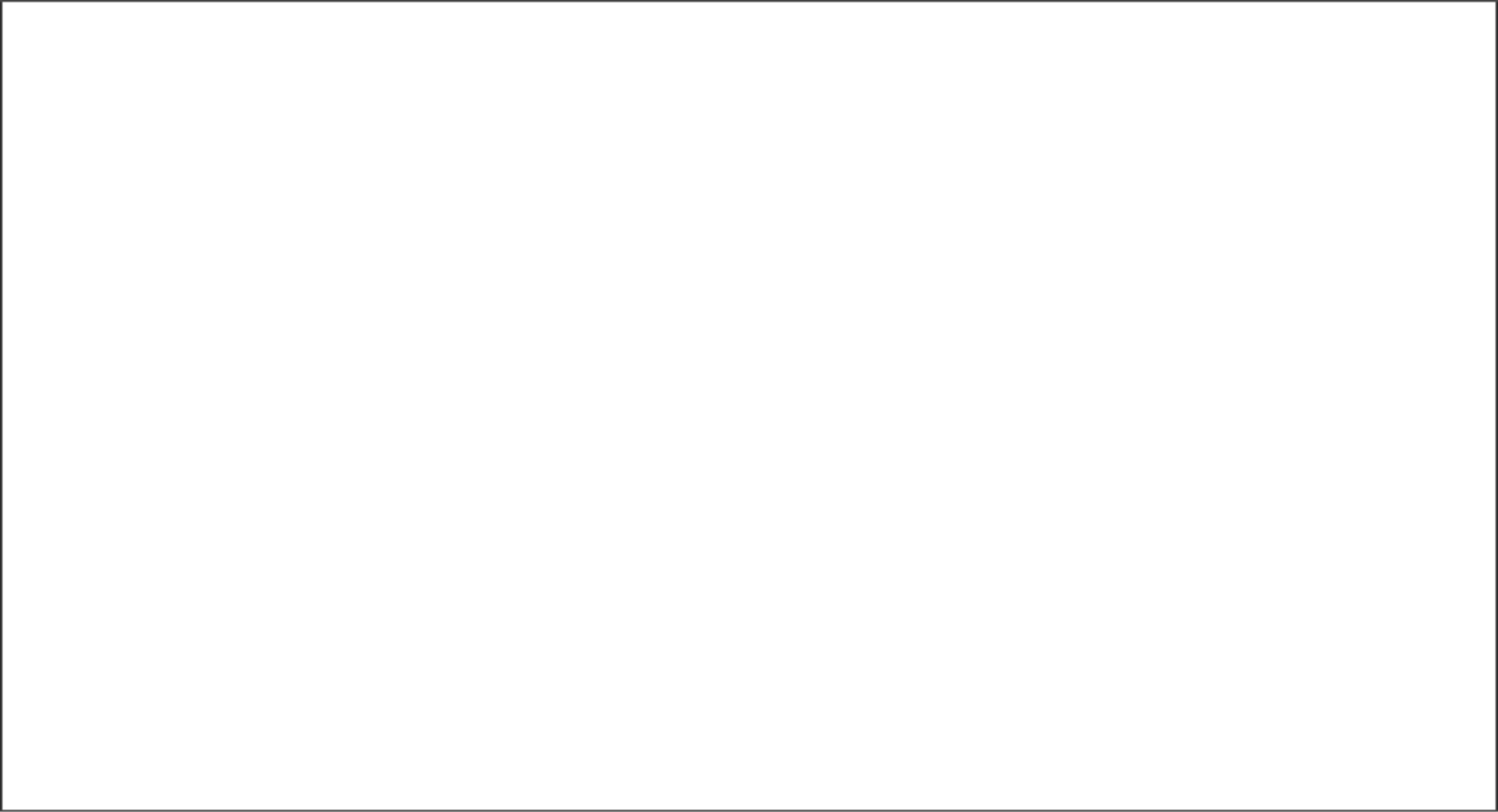
Por favor utilizar solo uno de los siguientes recursos en línea

- <http://creately.com/Draw-UML-and-Class-Diagrams-Online> (Entity Relationship UML)
- <https://www.gliffy.com/examples/flowcharts>
- <https://cacoo.com/signin>
- <https://online.visual-paradigm.com/es/solutions/free-flowchart-maker/>
- https://www.goconqr.com/es/users/sign_up

DISEÑO:

2.4. Por cada requerimiento funcional desarrollado en el numeral 1.2, crea un diagrama de flujo:

- <https://www.gliffy.com/examples/flowcharts>



3. WORKFLOW DE IMPLEMENTACIÓN:

3. Por cada requerimiento funcional desarrollado en el numeral 1.2, programa en un lenguaje de programación (consulta en referencia las pág. 34-39)

| | |
|---|---|
| Nombre: | { |
| Parámetros: | |
| Retorno: | |
| Descripción: | |
| } | |
| (Ejemplo) | { |
| Nombre: trabajo1 | |
| Parámetros: el modelo del vehículo | |
| Retorno: mensaje indicando el valor del servicio. | |
| Descripción: Función encargada de indicar la usuario el valor del servicio dependiendo del modelo del vehículo. | if(mod<2010) cout<<"S1. Alineación modelo anterior a 2010: \$80.000"<<endl; else cout<<"S1. Alineación modelo posterior a 2010: \$150.000"<<endl; } |
| Nombre: | |
| Parámetros: | |
| Retorno: | |
| Descripción: | |

| | |
|---------|--|
| Nombre: | |
|---------|--|

| | |
|--------------|--|
| Parámetros: | |
| Retorno: | |
| Descripción: | |
| Nombre: | |
| Parámetros: | |
| Retorno: | |
| Descripción: | |
| Nombre: | |
| Parámetros: | |
| Retorno: | |
| Descripción: | |

a. Código Fuente:

```
#include <iostream>
using namespace std;

main()
{
    char opcion;

    do
    {
        cout << " Bienvenido a mi programa\n";
        cout << " =====\n";
        cout << " 1. Requerimiento funcional 1\n";
        cout << " 2. Requerimiento funcional 2\n";
        cout << " 3. Requerimiento funcional 3\n";
        cout << " 4. Requerimiento funcional 4\n";
        cout << " 5. Exit\n";
        cout << " =====\n";
        cout << " Ingresa una opcion: "; cin
        >> opcion;
        cout << endl;

        switch (opcion)
        {
            case '1':
                cout << " Requerimiento funcional 1 \n ";
                cout << "\n";
                break;

            case '2':
                cout << " Requerimiento funcional 2 \n";
                cout << "\n";
                break;

            default:
                cout <<opcion << "is not a valid menu item.\n";
                cout << endl;
        }

    }while (opcion != '5' );
}
```

4. WORKFLOW DE PRUEBAS (TESTING):

- a. Seleccione dos atributos que sea solicitado al usuario de alguna de las estructuras del proyecto e ingrese un valor por fuera del rango establecido en el numeral 2.2., proceda a efectuar una captura con el resultado:

1:

2:

b. Realice una captura de dos casos de uso implementados en ejecución:

1:

2:

5. DOCUMENTACIÓN

- a. Documenta un manual de usuario:
 - i. Página de portada.
 - ii. Una página de título.
 - iii. Llenar el formato F277-Dirección Nacional de Derechos de Autor.pdf
[Solicitud inscripción soporte lógico \(software\)](#)
 - iv. Un prefacio, que contiene detalles de los documentos relacionados y la información sobre cómo navegar por la guía del usuario.
 - v. Una sección de introducción, que incluye:
 - a. Una breve descripción del sistema y su finalidad.
 - b. Una sección de novedades desde la última versión. (Anexar la bitácora con sus respectivas observaciones)
 - vi. Una sección de requisitos previos necesarios para usar el sistema, que incluye:
 - a. Conocimientos mínimos del usuario
 - b. Requisitos técnicos previos, incluyendo:
 - i. Capacidades técnicas mínimas del equipo
 - ii. Software asociado necesario
 - c. Mecanismo para acceder al sistema
 - vii. Una sección de instalación y configuración (si aplica)
 - viii. Una guía sobre cómo utilizar al menos las principales funciones del sistema, es decir, sus funciones básicas (requerimientos funcionales)
 - ix. Una sección de preguntas frecuentes. (FAQ)
 - x. Una sección de documentación técnica con el diseño de software
 - a. Diagrama entidad relación,
 - b. Diagrama de casos de uso,
 - c. Diagrama de clases.
 - xi. Datos de contacto de los autores.

- b. Entrega el programa con la última versión generada. Verifica que durante la ejecución lo primero que se presente sea los nombres de los autores del programa, tal como la portada de un documento. (Utiliza la instrucción `system("cls");`).

BIBLIOGRAFÍA

Ingresa a la Universidad Virtual:

Para descargar:
Formatos

Bibliotecas:
Biblioteca virtual y recursos
electrónicos

A continuación usted podrá hacer uso de las siguientes bibliotecas electronicas que la universidad tiene a su disposición

- [ebrary](#)
- [e-libro](#)
- [Cengage ebrary](#)
- [Gale Virtual Reference Library](#)
- [ProQuest](#)
- [Cengage](#)
- [Pearson](#)

Seleccione Pearson:

- Autor: Villalobos

www.pearsonbv.com/default.asp

PEARSON

INICIO

Biblioteca Virtual
de Contenidos Digitales

Regresar

Este título pertenece a: TEST • TEST1

Fundamentos de Programación

Aprendizaje activo basado en casos

Jorge A. Villalobos

No. Edición: 1

Año: 2006

ISBN e-Book: 9786073204552

ISBN: 9789702608462

Sinopsis

Tabla de Contenido

Ejemplares en la Biblioteca: 5