

TAREA N°1

Paradigma de la programación



INTEGRANTES:

1. VICENTE ROA
2. NICOLAS ORTIZ
3. BENJAMIN QUEZADA
4. AMAYA QUIROZ

INDICE

INTRODUCCIÓN	3
DESCRIPCIÓN DE LA PROBLEMÁTICA	3
DESCRIPCIÓN DE LA SOLUCION	3
Descripción del modelo.	4
CONCLUSIÓN	9

INTRODUCCIÓN

Sabemos que las evaluaciones son importantes para la educación ya que permite a los docentes conocer el nivel de sus estudiantes y así poder adaptar de mejor manera una enseñanza más efectiva para sus alumnos.

La Taxonomía de Bloom es una herramienta educativa ampliamente utilizada para clasificar diferentes niveles de habilidades y conocimientos en una jerarquía estructurada. En este proyecto, hemos construido un sistema que evalúa la comprensión del usuario en diferentes niveles de habilidades. En esta tarea, se implementó la Taxonomía de Bloom utilizando el lenguaje de programación C++.

El objetivo principal de este proyecto es comprender cómo se puede aplicar la Taxonomía de Bloom para desarrollar un sistema que evalúe el conocimiento de los estudiantes en diferentes niveles de habilidades.

DESCRIPCIÓN DE LA PROBLEMÁTICA

El programa de prueba consta de diferentes preguntas que se dividen en niveles de habilidades basados en la Taxonomía de Bloom. Cada pregunta se evalúa en función de su nivel de dificultad y se asigna a un nivel de habilidad específico.

Cuando el usuario responde una pregunta, el programa evalúa su respuesta y le proporciona una puntuación basada en el nivel de habilidad asignado a esa pregunta.

DESCRIPCIÓN DE LA SOLUCION

La taxonomía de Bloom es un marco de referencia utilizado para clasificar los niveles cognitivos que se espera que los estudiantes alcancen en su proceso de aprendizaje.

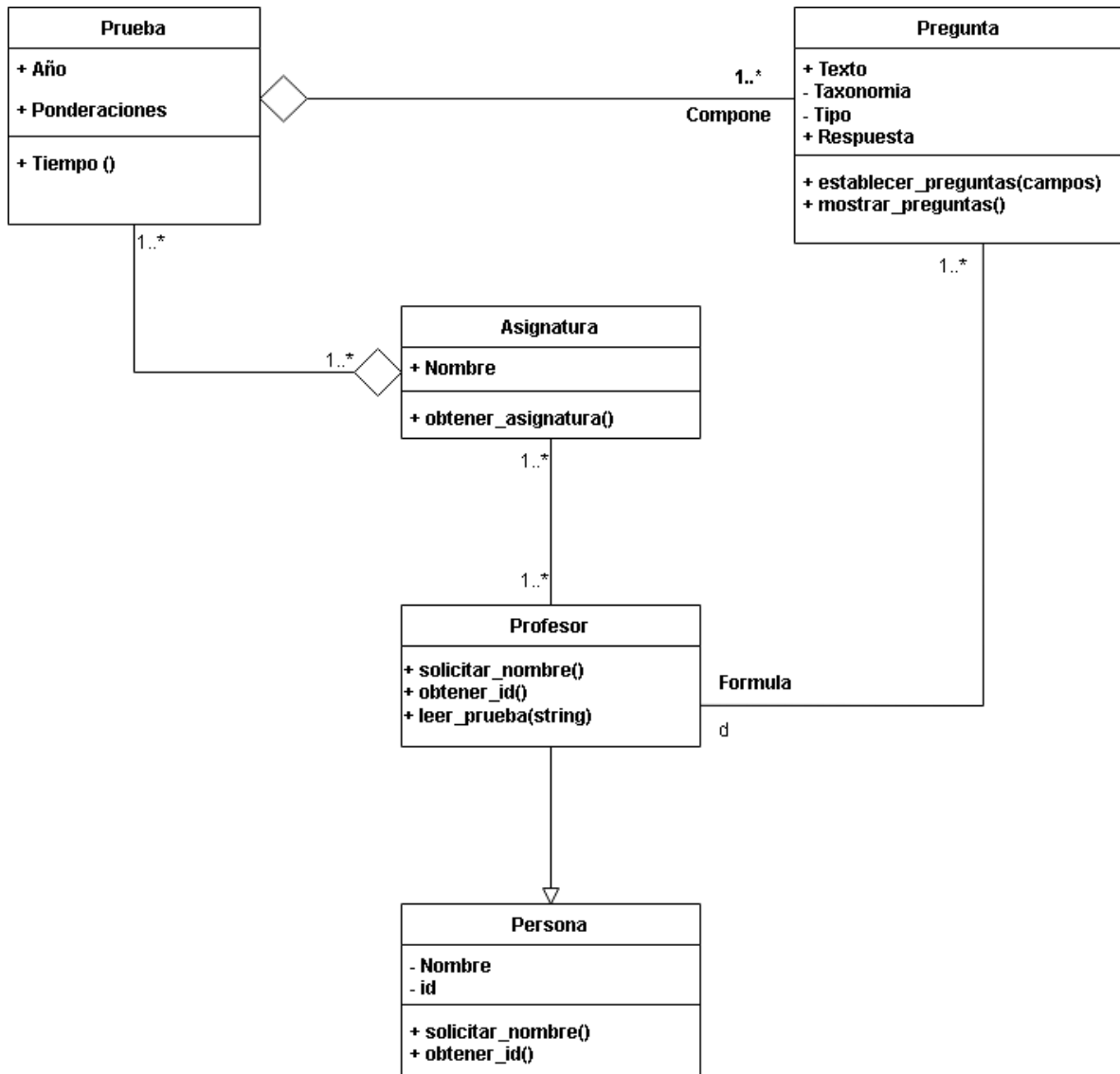
Esta taxonomía consta de seis niveles, que van desde lo más sencillo a lo más complejo, y cada nivel depende de los conocimientos adquiridos en los niveles anteriores.

A pesar de que la taxonomía de Bloom ha sido ampliamente utilizada en todo el mundo como una referencia para el diseño de pruebas y el desarrollo curricular, existen dificultades al aplicarla, especialmente en la enseñanza de la programación. Los motivos de estas dificultades son la dificultad para comprender las tareas cognitivas descritas en la taxonomía en el dominio de la programación, la falta de ejemplos en programación y la implicancia del investigador en el contexto de enseñanza del curso en donde se aplicó la evaluación. Sin embargo, a pesar de estos desafíos, la taxonomía de Bloom sigue siendo útil para especificar objetivos para que sea más fácil planificar experiencias de aprendizaje y preparar dispositivos de evaluación.

Descripción del modelo.

La solución que logramos como equipo fue.

Creación de un diagrama de clases: Una instancia que define muy bien nuestra problemática, nuestra solución y hacia donde queremos llegar con esta.



Para este diagrama creamos las clases Persona, Profesor, Asignatura, Prueba y Pregunta con sus atributos y métodos correspondientes.

Para esto relacionamos Persona como superclase sobre Profesor como subclase, luego desde profesor se relaciona con Pregunta con la cardinalidad de 1 a muchos.

Relacionamos profesor a asignatura, con una cardinalidad de muchos a muchos.

Luego, de la clase Asignatura se relaciona con una relación de agregación hacia prueba con una cardinalidad de muchos a muchos y por ultimo la clase Prueba se

esta relacionando de la forma agregación con la clase Pregunta con una cardinalidad de 1 a muchos.

Lluvia de ideas: Con esto logramos ponernos todos de acuerdo para llegar a un consenso grupal, esta lluvia de ideas nos ayudó para tener en cuenta como hacer el programa basado en C++, así como para pensar sobre una posible solución para esta problemática planteada anteriormente.

Creación del Programa.

Para la creación del programa creamos todas las clases correspondientes explicadas anteriormente en el diagrama de clases.

Creación de clases con sus constructores correspondientes.

1. Clase Prueba

```
// CLASE PRUEBA
class Prueba // Creamos la clase prueba
{
public:
    int anio;
    float ponderaciones;

    Prueba(int anio, float ponderaciones) // Constructor de la clase
    {
        this->anio = anio;
        this->ponderaciones = ponderaciones;
    }

    int tiempo() // Funcion principal de la clase prueba
    {
        cout << "\nTienes 10 min para realizar la prueba" << endl;
        return 0;
    }
};
```

2. Clase Pregunta

```
// CLASE PREGUNTA
class Pregunta {
public:
    string texto;
    string taxonomia;
    string tipo;
    string respuesta;

    // Constructor de la clase
    Pregunta(string texto, string taxonomia, string tipo, string respuesta) {
        this->texto = texto;
        this->taxonomia = taxonomia;
        this->tipo = tipo;
        this->respuesta = respuesta; // inicializa la respuesta como vacía
    }
}
```

3. Clase Profesor

```
// CLASE PROFESOR
class Profesor : public Persona {
public:
    Profesor(string nombre, int id) : Persona(nombre, id) {}
```

4. Clase Asignatura

```
// CLASE ASIGNATURA
class Asignatura {
public:
    string nombre;

    Asignatura(string nombre) { this->nombre = nombre; }

    void obtener_asig() { cout << "Nombre de la asignatura: " << nombre << endl; }
};
```

5. Clase Persona

```
// CLASE PERSONA
class Persona {
private:
    string nombre;
    int id;
public:
    Persona(string nombre, int id) // Constructor de persona
    {
        this->nombre = nombre;
        this->id = cont_persona + 1; // se sumara 1 al id para poder indentificarlos
        cont_persona++;
    }

    void solicitar_nombre() // Funcion para solicitar el nombre
    {
        cout << "\nIngresa el nombre de tu Profesor: ";
        cin >> nombre;
        cout << "El nombre de tu profesor a sido ingresado correctamente " << endl;
    }

    int obtener_id() // Funcion para obtener la id
    {
        cout << "id del profesor:" << id << endl;
        return id;
    }
};
```

Creamos las funciones:

Establecer preguntas: Esta función tiene como objetivo organizar los parámetros del archivo csv para ordenarlos como columnas, con esto logramos referirnos a cada uno de los índices, luego de esto podemos referirnos a cada columna como campos[i], como, por ejemplo, la columna “Pregunta”, está definida como campos[0].

Mostrar pregunta: Esta función tiene como objetivo mostrar la pregunta que está en el archivo csv, esta pregunta esta llamada como “texto”, que es una variable de tipo “string”.

```
// Función para establecer la respuesta
void establecer_preguntas(vector<string> campos) {
    this->texto = campos[0];
    this->taxonomia = campos[2];
    this->tipo = campos[1];
    this->respuesta = campos[3];
}

void mostrar_pregunta() { cout << texto << " " << endl; }
};

// nuestro contador de ID
int cont_persona = 0;
// CLASE PERSONA
class Persona {
private:
    string nombre;
    int id;
```

Colocamos la función `main()` llamando a todas las funciones anteriormente mencionadas y con esto finalizamos todo lo que es la parte de programación de nuestra problemática explicada anteriormente.

```
int main() {
    bool continuar = true;
    string respuesta, seleccion;
    int seleccion_preguntas;
    int contador = 0;

    menu();
    tiempo();
    Profesor profe("", 0);

    profe.solicitar_nombre();
    cout << "\n\n";
    system("pause");
    system("cls");

    /*PREGUNTAMOS LA CANTIDAD DE PREGUNTAS QUE QUIERE EL USUARIO*/
    cout << "\nIngrese la cantidad de preguntas: ";
    cin >> seleccion_preguntas;

    while (seleccion_preguntas > 0) {
        cout << "-----\n";
        while (contador < seleccion_preguntas) {
            fflush(stdin);
            cout << "Ingrese taxonomia: ";
            cin >> seleccion;
            profe.leer_prueba(seleccion);
            fflush(stdin);
            cout << "\n";
            cout << "Ingrese respuesta: ";
            cin >> respuesta;
            cout << "-----\n";
            contador++;
        }
        seleccion_preguntas--;
    }
    return 0;
}
```


CONCLUSIÓN

Para concluir, el programa de prueba tiene como objetivo evaluar y medir el nivel de conocimiento y habilidades del usuario en diversas áreas temáticas.

Para ello se utiliza un método basado en la clasificación de Bloom, que clasifica las preguntas en diferentes niveles de habilidad según su complejidad cognitiva. Así, cada pregunta del programa de prueba se puntúa según el nivel de dificultad y pertenece a un nivel específico de la taxonomía de Bloom. A medida que responde cada pregunta, el programa evalúa las respuestas del usuario y le otorga una puntuación que refleja su nivel de habilidad en relación con el nivel especificado para la pregunta.

Por lo tanto, el programa de prueba le permite identificar las fortalezas y debilidades de los usuarios en cada área temática y brinda información útil para mejorar las habilidades en aquellas áreas donde se requiere trabajo adicional.

Esto hace que el programa de evaluación sea una herramienta valiosa tanto para usuarios individuales como para instituciones educativas o comerciales que buscan evaluar y mejorar las habilidades de los empleados.