

# 0.0 Condiciones

## Límite de Entrega

Antes del viernes 2 de marzo de 2007 a las 11:30

## Formato de entrega

- Dos ficheros con la declaración de las clases:

Alumno.h

Alumno\_M2.h

- Dos ficheros con la implementación de las clases:

Alumno.cpp

Alumno\_M2.cpp

(\*) Los ficheros deben llevar la cabecera explicada como estándar en clase con el nombre y apellidos del alumno autor

## Forma de entrega

Correo:

[jmarina@nebrija.es](mailto:jmarina@nebrija.es)

# 1.0 Definición

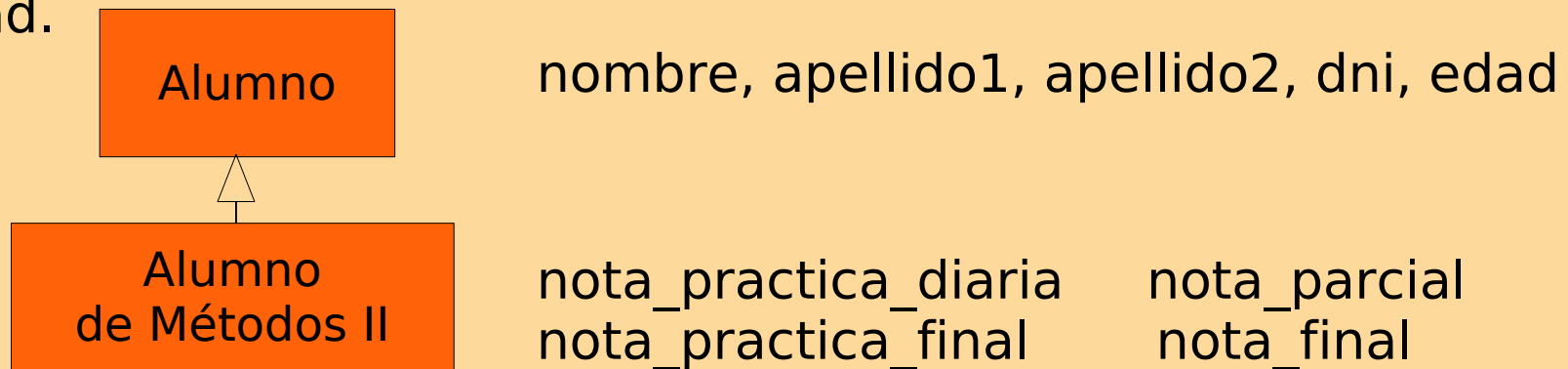
## Definición

Queremos poder calcular la nota de los alumnos de la asignatura de Métodos II de programación sabiendo que la nota final se calcula:

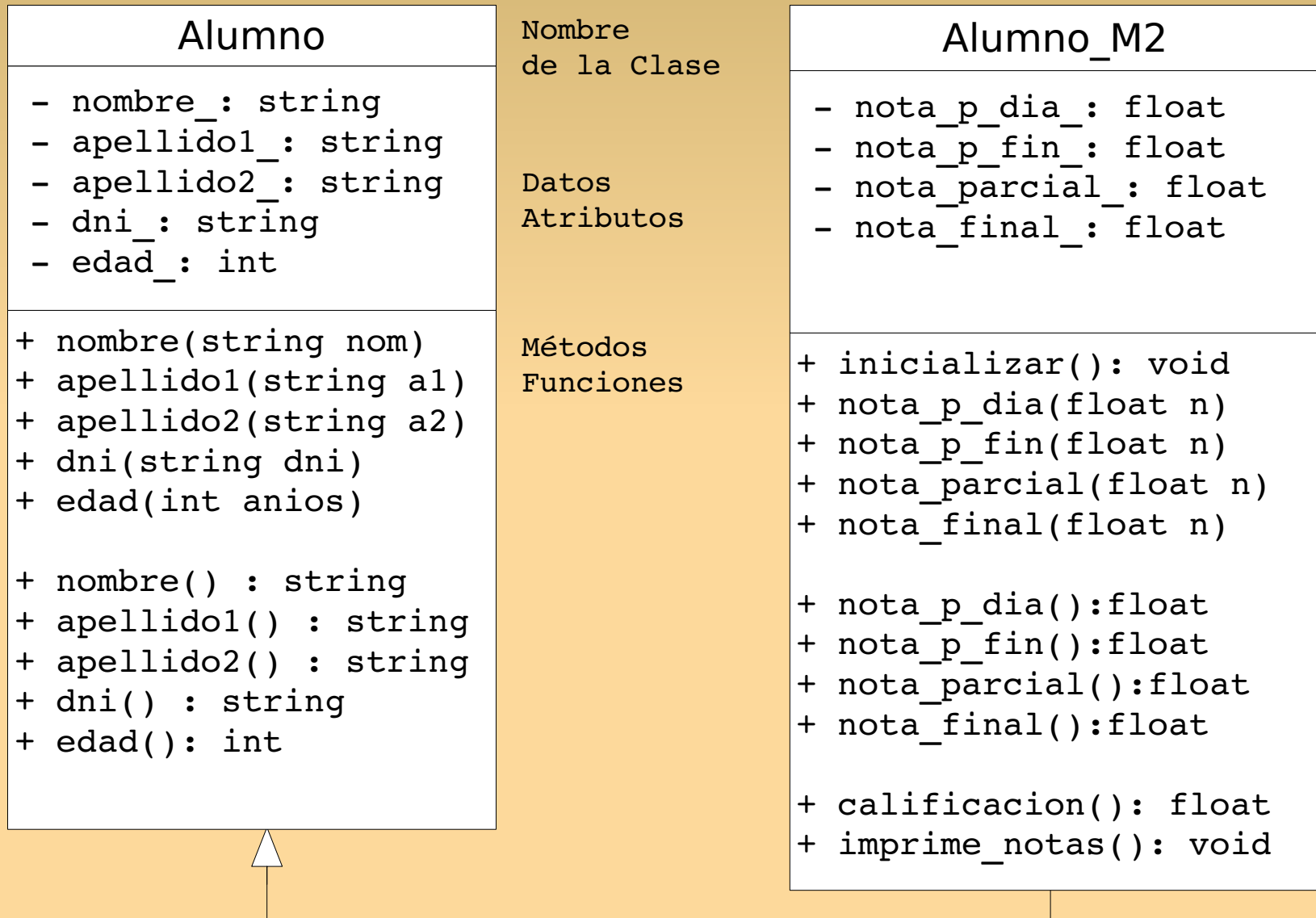
- 20 % Por las prácticas de laboratorio (20% diarias 80% trabajo final)
- 15 % Por examen parcial.
- 65 % Por examen final

y suponiendo que todos los alumnos aprobarán y no irán al examen extraordinario.

Los alumnos deberán ser identificados por nombre, apellido1, apellido2 y DNI y edad.



# 1.2 Diagrama UML



# 1.3 Ayuda

## Definición de Interfaz

```
// Clase Alumno de Métodos II
class AlumnoM2: public Alumno
{
private:
    float  nota_p_dia_;
    float  nota_p_fin_;
    float  nota_parcial_;
    float  nota_final_;

public:
    void inicializar (void);
    void nota_p_dia (float nota);
    void nota_p_fin (float nota);
    void nota_parcial (float nota);
    void nota_final (float nota);

    float nota_p_dia (void);
    float nota_p_fin (void);
    float nota_parcial (void);
    float nota_final (void);

    float calificacion(void);
    void imprime_notas();
}
```

### AlumnoM2

```
- nota_p_dia_: float
- nota_p_fin_: float
- nota_parcial_: float
- nota_final_: float
```

```
+ inicializar(): void
+ nota_p_dia(float n)
+ nota_p_fin(float n)
+ nota_parcial(float n)
+ nota_final(float n)
```

```
+ nota_p_dia():float
+ nota_p_fin():float
+ nota_parcial():float
+ nota_final():float
```

```
+ calificacion(): float
+ imprime_notas(): void
```

# 1.3 Ayuda

## Implementación

```
void
AlumnoM2::inicializar (void)
{
    nota_p_dia_      = -1;
    nota_p_fin_      = -1;
    nota_parcial_    = -1;
    nota_final_      = -1;
}
//-----
void
AlumnoM2::nota_p_dia (float nota);
{
    nota_p_dia_      = nota;
}
//-----
void
AlumnoM2::nota_p_fin (float nota);
{
    nota_p_fin_      = nota;
}
//-----
void
AlumnoM2::nota_parcial (float nota);
{
    nota_parcial_    = nota;
}
```

```
//-----
void
AlumnoM2::nota_final (float nota);
{
    nota_final_      = nota;
}
//-----
float
AlumnoM2::nota_final (void);
{
    return nota_final_;
}
...
//-----
float
AlumnoM2::calificacion (void);
{
    float cal=0.0;
    cal = 0.2* (0.2 * nota_p_dia_ +
                0.8 * nota_p_fin_) +
          0.15*nota_parcial_ +
          0.65*nota_final_;
    return cal;
}
```

# 1.4 Ejercicio

0.- Realizar declaración e implementación de las clases Alumno y Alumno\_M2.

Codificar en ficheros separados cada clase y cada clase con su declaración en un .h y su implementación en un .cpp.

Deben ir adecuadamente comentados:

- Fichero, Clases, Funciones, Parámetros y valor de retorno.

1.- Método para imprimir

Codificar el método de la clase Alumno\_M2 para imprimir por pantalla:

```
Nombre      :  
Apellido1   :  
Apellido2   :  
DNI         :  
Edad        :
```

```
Calificación :
```

2.- Método que devuelve la calificación en un argumento.

Codificar el método que devuelve en un argumento en lugar de como valor de retorno.