

Cómo congeniar los exámenes y los proyectos en asignaturas PBL

Pablo del Canto, Isabel Gallego, Rubén Hidalgo, Johann López, José Manuel López, Javier Mora, **Eva Rodríguez**, Eduard Santamaria, Miguel Valero

Escola Politècnica Superior Castelldefels
Universitat Politècnica Catalunya



Contenido

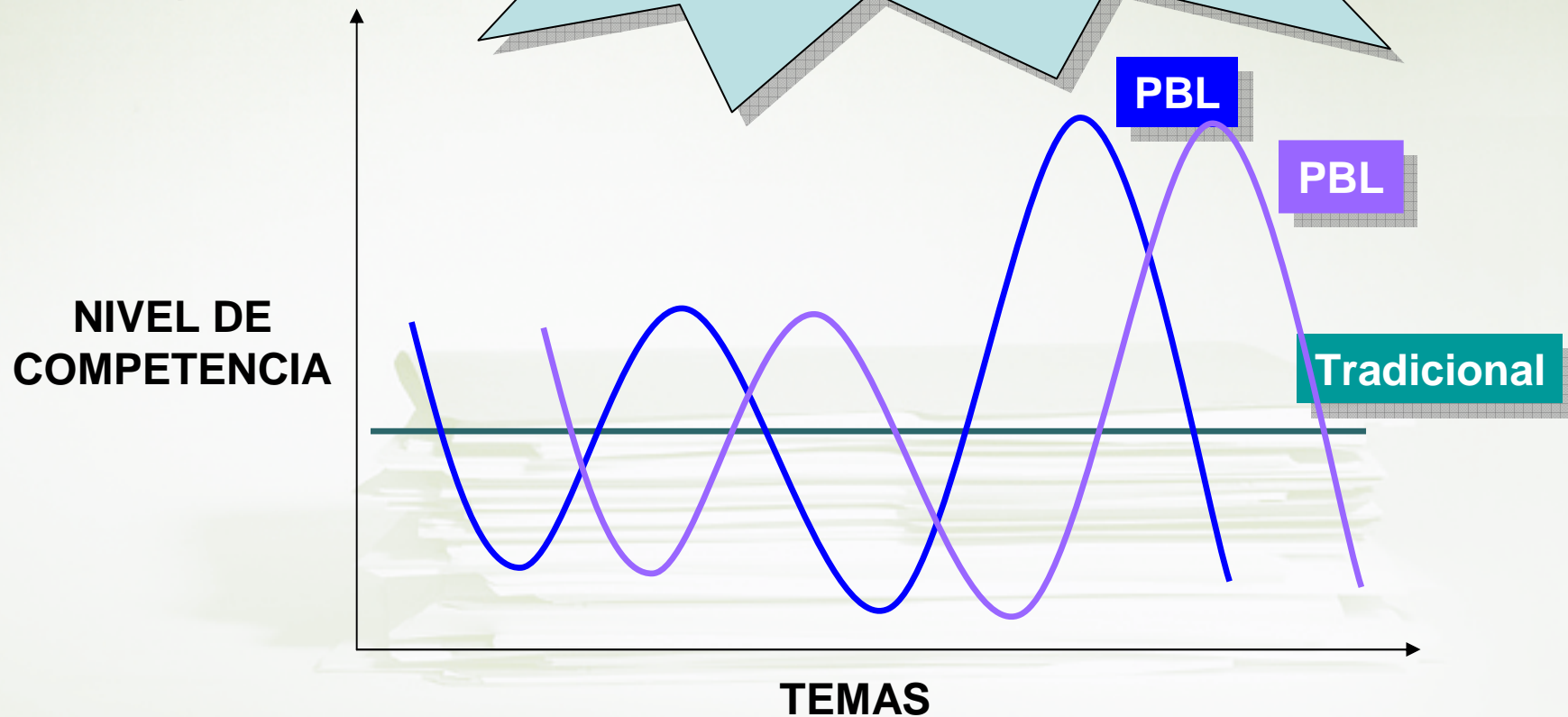
- Contexto
- Aprendizaje individual y el PBL
- Los Conocimientos Mínimos
 - Cómo identificar los CMs?
 - Los CMs de IC
 - Cómo se presentan los CMs a los alumnos?
 - Consideraciones sobre la calificación de los CMs
- La Evaluación y el PBL
 - El método de evaluación de IC
- Resultados obtenidos
- Conclusiones

Contexto

- EPSC-DAC
 - Ingeniería Técnica de Telecomunicación
 - Ingeniería Técnica Aeronáutica
- Fase selectiva: Primer curso
 - IC: Programación C (1A)
 - LP-TCP: Programación C++ & VBasic (1B)
- Créditos ECTS: 4.6
- Trabajo personal: 8 horas/semana
 - 2 (4) horas sesión presencial
 - 6 (4) horas trabajo personal individual o en grupo
- Orientadas al PBL
 - IC: Segunda mitad del cuatrimestre
 - LP-TCP: Todo el cuatrimestre

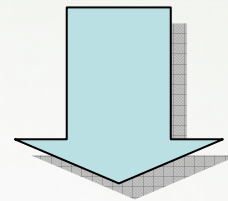
Aprendizaje individual & PBL

Menor cobertura en algunos temas,
a cambio de mayor profundidad en otros



La evaluación y el PBL

Nos sorprenden con sus proyectos
Nos decepcionan con sus exámenes

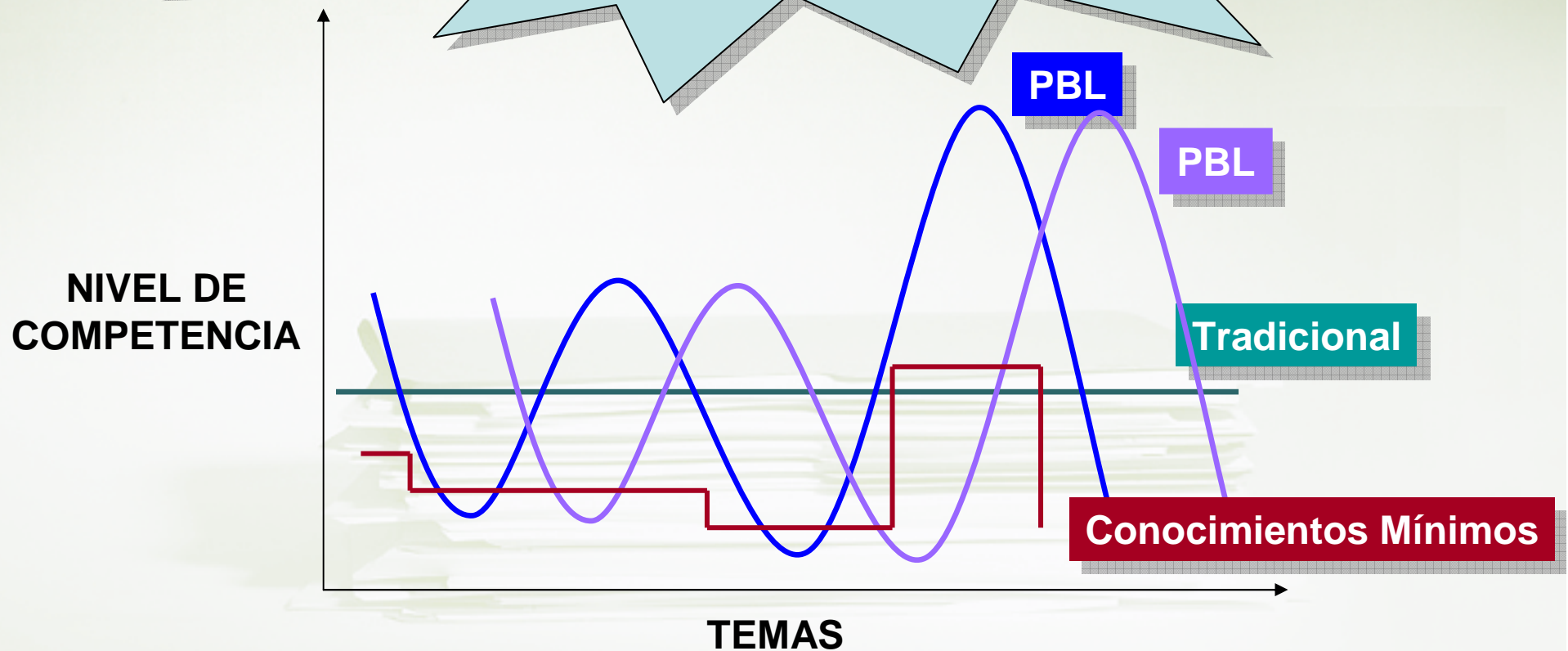


Coexistir Exámenes con Proyectos?

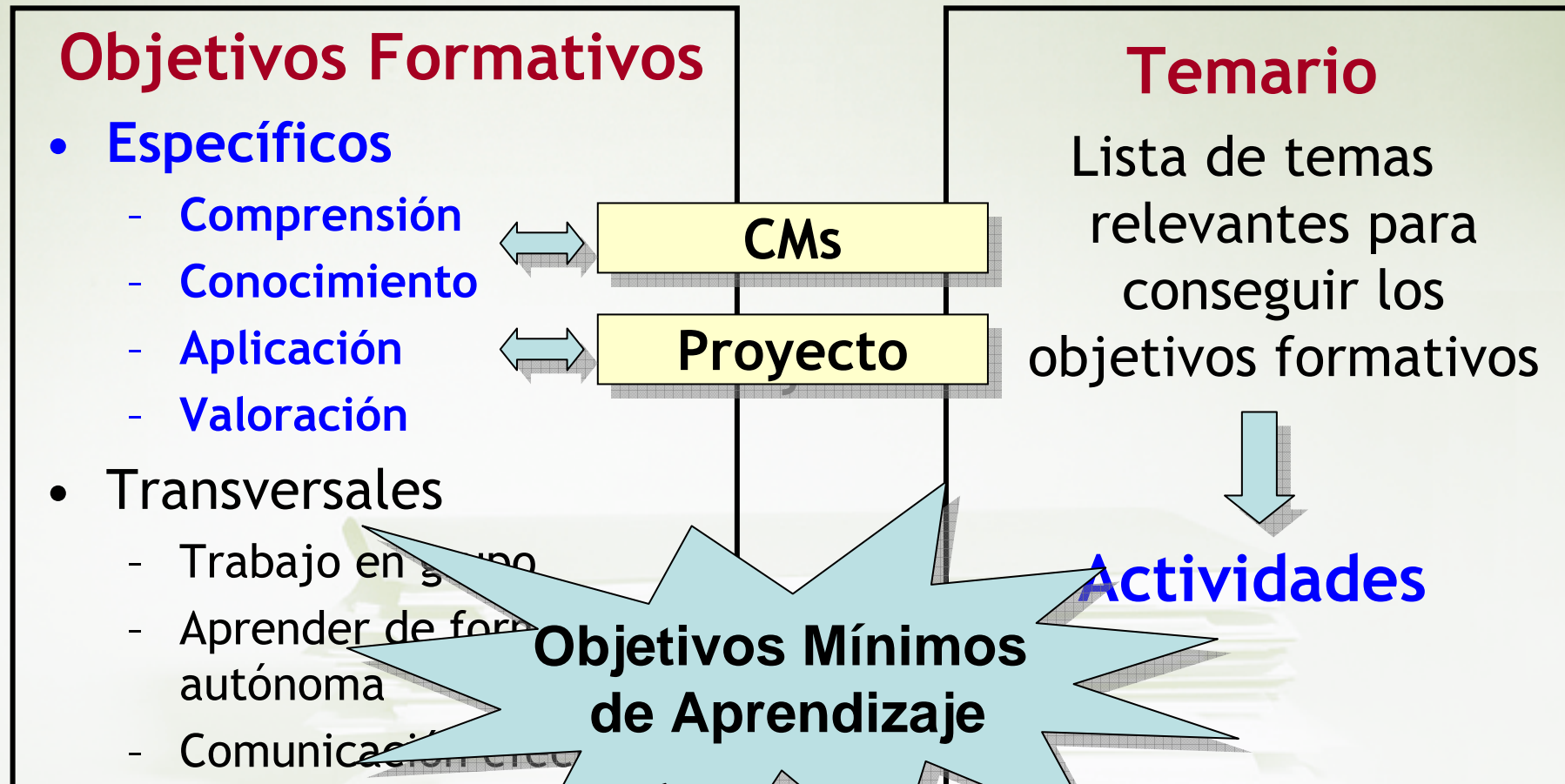
Conocimientos Mínimos

Aprendizaje individual & PBL

Menor cobertura en algunos temas,
a cambio de mayor profundidad en otros



Cómo identificar los CMs?



Los Conocimientos Mínimos de IC

Conocimiento Mínimo	Definición
1	Escribir un programa que lea información de línea de comandos (caracteres, enteros, reales o palabras), tome decisiones a partir de esa información (usando solamente sentencias condicionales) y escriba en la pantalla los resultados (enteros, reales, palabras, frases)
2	Escribir un programa que realice un recorrido o una búsqueda en una secuencia de dígitos, números consecutivos o números leídos del teclado
3	Escribir un programa que haga un recorrido o una búsqueda en un vector o matriz de caracteres o números
4	Escribir un programa que requiera la combinación de dos esquemas algorítmicos (recorrido y/o búsqueda) aplicados sobre un vector o matriz de enteros o caracteres
5	Diseñar la estructura de datos más adecuada para representar la información de un problema dado
6	Codificar un procedimiento o función que cargue una estructura de datos con la información contenida en un fichero de texto o que salve en un fichero de texto la información contenida en la estructura
7	Escribir un procedimiento o función que recorra, busque o añada información en un vector de estructuras
8	Escribir un programa principal que realice llamadas a procedimientos o funciones pasándole adecuadamente los parámetros, leyendo del teclado la información necesaria para realizar las llamadas y escribiendo en pantalla la información correspondiente a los resultados de las llamadas

Como se presentan los CM a los alumnos?

Conocimientos mínimos – Ejemplos con sus soluciones

Conocimiento mínimo 2

Escribir un programa que realice un recorrido o una búsqueda en una secuencia de dígitos, números consecutivos o números leídos del teclado.

Ejemplos

Solución

2.1 Realizar un programa que pida un número entero positivo por teclado y diga cuál es su mayor dígito par y su menor dígito impar.

[solución](#)

2.2 Realizar un programa que determine la suma de los 10 primeros múltiplos de 6.

[solución](#)

2.3 Identificar el mayor de los tres primeros múltiplos de 7 en la secuencia de números escritos por teclados, acabada en 0.

[solución](#)

Conocimiento mínimo 4

Escribir un programa que requiera la combinación de dos esquemas algorítmicos (recorrido y/o búsqueda) aplicados sobre un vector o matriz de enteros o caracteres.

Ejemplos

Solución

4.1. Escribir un programa que lea del teclado un número entero y lo inserte en un vector de enteros que está ordenado de menor a mayor, de forma que el vector se mantenga ordenado de menor a mayor.

[solución](#)

4.2 Suponer las siguientes declaraciones:

```
#define MAX 20
```

```
int vector [MAX];
```

Suponer que el vector ya contiene números enteros en todas las posiciones. Escribir un programa que elimine del vector el primer múltiplo de 7 que encuentre. Para eliminar un número hay que mover todos los que tiene a su derecha una posición a la izquierda.

[solución](#)

4.3 Suponer las siguientes declaraciones:


```
#define MAX 20
```

```
char frase [MAX];
```

Suponer que el vector frase ya contiene caracteres en todas las posiciones. Escribir un programa que inserte un carácter en blanco después de la primera 'j' que encuentre

[solución](#)

Plan de actividades



UNIVERSITAT POLITÈCNICA DE CATALUNYA

Plataforma de suport a la docència **AZENEÀ** v4.3

Atenea 4.3 » EPSC-5003-Tota4 » Recursos » Semana 6: Guía de las actividades

Actualiza aquest Recurs

Semana 6: Guía para las actividades

- 1. Sesión de clase S6 (2h)**

Entregable de grupo temporal (#6.1): Ejercicios resueltos
- 2. Ejercicios individuales sobre vectores (2 h)**
- 3. Puesta en común en grupos base (2h).**

Entregable de grupo (#6.2): Soluciones acordadas para los ejercicios
- 4. Ensayo de primera prueba de mínimos y autoevaluación (2 h).**

Este es un ejemplo de enunciado de primera prueba de mínimos como la que tendrás la semana 7. Intenta resolver los ejercicios sobre papel. Cuando los hayas resuelto, compara tu solución con [esta solución oficial](#), y decide si tu código es correcto (bien, regular o mal) y si es claro (bien, regular o mal) según los criterios siguientes:

Código correcto:

Bien : El código es prácticamente igual a la solución. Hay alguna pequeña diferencia con la solución, pero puedo justificarla perfectamente.

Regular: El código es similar a la solución. Hay uno o dos errores en el código que se pueden arreglar sin tener que empezar de nuevo.

Mal : Hay muchos errores. Para arreglar mi solución debería empezar de nuevo.

Código claro:

Bien : El código está bien indentado, y hay un criterio coherente para usar los {}. Los comentarios ayudan a identificar claramente los componentes del esquema que se ha aplicado.

Regular : Hay alguna parte del código que no está bien indentada, y algún comentario no es suficientemente claro.

Mal : El código no está indentado, y no hay comentario, o los que hay no aclaran nada.

Entregable individual (#6.3): Informe de autoevaluación, que debes entregar en el Campus Digital.
- 5. Preparación de dudas (15')**

Entregable individual (#6.4): Hoja con las dudas

El método de evaluación de IC (1/2)

Elemento	Peso
Entregables del curso	20%
Conocimientos mínimos	30%
Proyecto Versión 1	10%
Proyecto Versión 2	20%
Ejercicio individual de ampliación del Proyecto	10%
Actitud y participación	10%

El método de evaluación de IC (2/2)

- Entregables: **20%**
 - Mínimo 80% presentados (Grupo e Individuales)
- Proyecto: **40%**
 - Primer prototipo (Grupo)
 - Versión final (Grupo)
 - Ejercicio de ampliación (Individual)
- **Conocimientos Mínimos: 30%**
 - **Mínimo 7 de 8**
- Actitud & Participación: **10%**

Consideraciones sobre la calificación de los CMs

- Los CMs se superan o no se superan
- No se da una calificación de 0 a 10
- Se reduce considerablemente el tiempo de corrección
- Papel
 - Pequeño error superable con un PC y más tiempo
- Ordenador
 - Funciona / No funciona

Coincidència	%
4 de 4	68,81
3 de 4	23,65
2 de 4	7,52

Calificación de los CMs

10	Si demuestran todos los conocimientos mínimos sin necesidad de utilizar la convocatoria de la semana de exámenes finales
8,5	Si demuestran todos los conocimientos, pero necesitan la convocatoria de la semana de exámenes finales
7,5	Si demuestran todos los conocimientos menos uno sin necesidad de la convocatoria de la semana de exámenes finales
5	Si demuestran todos los conocimientos mínimos menos uno, usando la convocatoria de la semana de exámenes finales
4,5	Si faltan dos mínimos pero la calificación del proyecto es mayor que 7
0 - 3,5	En función del número de conocimientos mínimos superados

Resultados obtenidos

IC 2006-07 Q1	%
Demostrado todos CMs a la primera	50%
Demostrado todos CMs con recuperación	24,48%
Demostrado todos los CMs menos 1 a la primera	6,12%
Demostrado todos los CMs menos 1 con recuperación	9,18%
Suspendidos: Han realizado bien el proyecto pero no han demostrado los CMs	10,2%

Conclusiones

- Congeniar exámenes y proyectos
- Método evaluación diseñado en función de la adquisición de unos **Conocimientos Mínimos**
- **Garantizar** que **todos** los alumnos **alcancen** los **objetivos mínimos** de **aprendizaje**
- CMs Están mejor o peor formados?
 - **No lo se!**
- Se reduce el tiempo de corrección

Cómo congeniar los exámenes y los proyectos en asignaturas PBL

Pablo del Canto, Isabel Gallego, Rubén Hidalgo, Johann López, José Manuel López, Javier Mora, **Eva Rodríguez**, Eduard Santamaria, Miguel Valero

Escola Politècnica Superior Castelldefels
Universitat Politècnica Catalunya