

# PROJECTE FINAL DE CARRERA

## MEMÒRIA FINAL

Títol: Cloud Code Test IDE

Estudiant: Joan Carbonell Picó

Consultor: Jordi Duran Cals



Universitat Oberta  
de Catalunya

# ÍNDIX

1.- INTRODUCCIÓ.....	7
1.1.- CONTEXT.....	7
1.2.- JUSTIFICACIÓ DEL PROJECTE.....	7
1.3.- OBJECTIUS.....	9
1.4.- ESTRUCTURA DE LA MEMÒRIA.....	11
2.- ESTUDI DE VIABILITAT.....	12
2.1.- PLANIFICACIÓ (DIAGRAMA DE GANTT).....	12
2.2.- FASES DEL PROJECTE.....	13
2.2.1.- FASE 1: Pla del project.....	13
2.2.2.- FASE 2: Tecnologies de desenvolupament IDE en el núvol.....	13
2.2.3.- FASE 3: Compilació i execució al núvol.....	14
2.2.4.- FASE 4: Arquitectura de l'aplicació Web.....	15
2.2.5.- FASE 5: Tecnologies aplicades.....	15
2.2.6.- FASE 6: Anàlisi de l'entorn web destinat a la programació en C/C++.....	17
2.2.7.- FASE 7: Disseny de l'entorn web destinat a la programació en C/C++.....	17
2.2.8.- FASE 8: Desenvolupament inicial de l'aplicació web.....	18
2.2.9.- FASE 9: Desenvolupament opcional de l'aplicació web.....	19
2.2.10.- FASE 10: Proves.....	20
2.2.11.- FASE 11: Memòria del projecte.....	20
2.2.12.- FASE 12: Presentació virtual.....	21
2.2.13.- FASE 13: Defensa del projecte.....	21
2.3.- ENTORN DE DESENVOLUPAMENT.....	22
2.4.- METODOLOGIA A SEGUIR.....	22
2.5.- RESULTAT FINAL.....	23
3.- DISSENY.....	24
3.1.- TECNOLOGIES DE DESENVOLUPAMENT IDE EN EL NÚVOL.....	24
3.1.1.- Introducció .....	24
3.1.2.- IDE's AL NÚVOL.....	25
SourceLair.....	25
Cloud9.....	25
Ideone.....	26
Compileonline.....	26
Codepad.....	26
3.1.3.- Espais d'un IDE al núvol.....	27
Espais-Alumnat.....	27
Espais-Professorat.....	28

3.1.4.- Compilació i execució als IDE's remots.....	28
3.1.5.- Conclusió.....	29
3.2.- COMPILACIÓ I EXECUCIÓ AL NÚVOL.....	29
3.2.1.- Introducció.....	29
3.2.2.- Emscripten.....	30
3.2.3.- Instal·lació.....	30
3.2.4.- Comprovació.....	32
3.2.5.- Compilació - Execució.....	32
Client-Servidor (execució al costat del servidor).....	32
Client-Servidor (execució al costat del servidor i del client).....	34
Client-Client (execució al costat del client).....	34
3.2.6.- Conclusions.....	36
3.3.- ARQUITECTURA DE L'APLICACIO WEB.....	37
3.3.1.- Introducció.....	37
3.3.2.- Funcionament de l'IDE.....	39
3.3.3.- Arquitectura client – servidor.....	39
3.3.4.- Patró Model-Vista-Controlador.....	41
3.4.- TECNOLOGIES APLICADES.....	42
3.4.1.- Introducció.....	42
3.4.2.- Tecnologies al costat del servidor.....	43
Node.js.....	43
Mòduls.....	44
npm .....	44
express.....	44
jade.....	44
socket.io.....	45
passport.....	45
mysql.....	45
nodemon.....	45
3.4.3.- Tecnologies al costat del client.....	45
JQuery.....	45
3.5.- ANÀLISI DE L'ENTORN WEB DESTINAT A LA PROGRAMACIÓ C++.....	46
3.5.1.- Introducció .....	46
3.5.2.- Requisits funcionals.....	46
Professor – Requisits funcionals.....	47
Alumne – Requisits funcionals.....	48

Actors – Casos d'ús.....	49
3.5.3.- Descripció dels casos d'ús.....	51
DESCRIPCIÓ TEXTUAL.....	52
Sistema genèric.....	52
Cas d'ús: Iniciar sessió.....	52
Cas d'ús: Tancar sessió.....	52
Cas d'ús: Llistar usuaris d'un grup.....	53
Cas d'ús: Llistar pràctiques.....	53
Cas d'ús: Visualitzar solució.....	53
Cas d'ús: Llistar grups.....	54
Sistema d'administració.....	54
Cas d'ús: Crear grup.....	54
Cas d'ús: Eliminar grup.....	55
Cas d'ús: Crear usuari.....	56
Cas d'ús: Eliminar usuari.....	56
Sistema docent.....	57
Gestionar pràctica.....	57
Cas d'ús: Crear pràctica.....	57
Cas d'ús: Afegir pràctica/grup.....	58
Cas d'ús: Extraure pràctica/grup.....	58
Cas d'ús: Eliminar pràctica.....	59
Cas d'ús: Modificar pràctica.....	59
Gestionar solució.....	60
Cas d'ús: Afegir Solució.....	60
Cas d'ús: Eliminar Solució.....	61
Cas d'ús: Modificar Solució.....	62
Cas d'ús: Publicar Solució.....	62
Gestionar prova.....	63
Cas d'ús: Afegir prova.....	63
Cas d'ús: Modificar prova.....	64
Cas d'ús: Eliminar prova.....	65
Sistema discent.....	66
Gestionar edició.....	66
Cas d'ús: Visualitzar pràctica.....	66
Cas d'ús: Crear treball.....	66
Cas d'ús: Visualitzar treballs.....	67

Cas d'ús: Obrir treball.....	67
Gestionar execució.....	68
Cas d'ús: Executar treball.....	68
Cas d'ús: Comprovar prova.....	69
Cas d'ús: Visualitzar resultat.....	70
Cas d'ús: Visualitzar solució.....	70
3.5.4.- Requisits no funcionals.....	70
3.6.- DISSENY DE L'ENTORN WEB DESTINAT A LA PROGRAMACIÓ EN C++.....	71
3.6.1.- Introducció .....	71
3.6.2.- Diagrama estàtic d'anàlisi.....	71
Usuari.....	71
Professor.....	72
Alumne.....	72
Administrador.....	72
Grup.....	72
Practica.....	72
Solució.....	72
Prova.....	72
Resultat.....	73
Treball.....	73
3.6.3.- Interfícies.....	74
Espai global.....	74
Espai alumne.....	75
Espai professor.....	76
Espai administrador.....	76
3.6.4.- Execució i compilació.....	78
4.- IMPLEMENTACIÓ.....	82
4.1.- DESENVOLUPAMENT INICIAL DE L'APLICACIÓ WEB.....	82
4.1.1.- Introducció.....	82
4.1.2.- Creació del lloc web.....	82
4.1.3.- Estructura del lloc web.....	83
4.1.4.- Aplicació bàsica.....	84
Rutes.....	85
Vistes.....	89
Espai de treball.....	89
4.2.- DESENVOLUPAMENT OPCIONAL DE L'APLICACIÓ WEB.....	89

---

4.2.1.- Introducció .....	89
4.2.2.- Creació de la base de dades .....	90
4.2.3.- Creació de les vistes definides a la fase anterior.....	91
Administrador.....	91
Professor.....	93
Alumne.....	97
4.2.4.- Implementació de la lògica.....	99
5.- LIMITACIONS I PROVES.....	99
5.1.- PROVES.....	99
6.- CONCLUSIONS.....	101
6.1.- RECAPITULACIÓ.....	101
6.2.- AMPLIACIONS.....	101
6.3.- VALORACIÓ PERSONAL.....	101
7.- REFERÈNCIES.....	102
Arquitectura.....	102
Emscriptem.....	102
LLVM.....	102
IDE al núvol.....	102
Node.js.....	102
Express.....	103
Jade.....	103
Apunts UOC.....	103
Mysql.....	103
C++.....	103

# 1.- INTRODUCCIÓ

## 1.1.- CONTEXT

Actualment, als estudis de secundària s'ofereix l'assignatura de tecnologies de la informació i la comunicació com una optativa que es pot cursar a qualsevol curs de l'ESO i batxillerat. Durant l'etapa obligatòria (ESO) la majoria dels continguts s'estructuren en els següents blocs temàtics: sistemes operatius i dispositius d'emmagatzematge, Internet i les eines col·laboratives, aplicacions ofimàtiques i multimèdia. Respecte als estudis postobligatoris del batxillerat, els continguts solen ser una continuació dels anteriors però adaptats a les seues capacitats.

Tot i això, al segon curs de batxillerat es contempla l'estudi d'un llenguatge de programació estructurat. Encara que el professorat és lliure d'elegir el llenguatge a estudiar, és comú que aquest siga C o C++. El principal motiu és perquè s'estudia als diferents cicles formatius de grau mitjà i superior de la família d'informàtica i també perquè sol formar part dels continguts que hi ha a les assignatures d'informàtica de moltes de les enginyeries que volen estudiar els alumnes. Evidentment, la llibertat de càtedra de cada professor fa possible que l'elecció d'aquest llenguatge siga per uns altres motius diferents als esposats anteriorment.

Com que cada alumne pot presentar unes característiques i situacions personals diferents, el sistema educatiu ofereix les modalitats presencials i a distància per a poder cursar estudis a l'ESO, batxillerat o cicles formatius. Per a l'estudi d'un llenguatge de programació, caldrà tenir en compte aquestes dues modalitats. Però a més també és important oferir eines apropiades per a donar resposta a les necessitats, temporals o permanents, de tot l'alumnat del centre. Pensem que un alumne matriculat en la modalitat presencial pot tenir una incapacitat temporal (una operació, per exemple) que li impedisca la seua assistència a l'aula d'informàtica. Per tant, l'atenció a la diversitat requerirà accions educatives que previnga i done solucions a les mancances de l'alumnat. A l'estudi del llenguatge de programació estructurada, el professorat haurà d'oferir les eines adequades per a les diferents situacions que se li poden presentar.

## 1.2.- JUSTIFICACIÓ DEL PROJECTE

L'alumnat de les diferents etapes educatives no universitàries que estudia un llenguatge de programació estructurat com C/C++ presenta un gran disposició per al seu aprenentatge. Una de les primeres tasques que haurà de fer és preparar l'entorn de desenvolupament per a poder estudiar aquest. Tot i això, els ordinadors de les aules dels centres de secundària ja incorporen l'entorn adient per aquesta finalitat i l'alumnat sol ha de començar a utilitzar-lo. En altre cas, el professor responsable de les aules s'encarregarà d'instal·lar-lo. Evidentment, als cicles formatius l'alumnat rep

aquesta formació, però a l'ESO i batxillerat normalment no.

Altra cosa a considerar és que la plataforma utilitzada per a desenvolupar la tasca docent a les aules d'informàtica és Lliurex (edubuntu). En canvi l'alumnat sol treballar amb els seus ordinadors personals amb el sistema operatiu Windows. Encara que molts d'ells saben instal·lar l'entorn per a desenvolupar programes en C/C++, altres no. També hem tenir en compte altres situacions que es poden donar a les famílies dels alumnes: no tenen ordinador a casa, el que tenen no funciona i no tenen coneixements per a posar-lo operatiu de nou, utilitzen els ordinadors de les biblioteques o dels casals de joventut, etc.. Totes aquestes qüestions impossibiliten que l'alumnat pugui seguir l'assignatura davant de qualsevol imprevist.

També hem de tenir en compte a l'alumnat que estudia a distància un llenguatge de programació. Aquests seguiran les indicacions del professorat per qualsevol mitjà dels disponibles a Internet, però molts d'ells requeriran de més interacció amb el professor per poder conèixer la seua evolució. Per exemple, si un alumne vol saber si el programa implementat obté els resultats esperats, haurà de rebre els arxius que testegen el bon funcionament del programa via correu electrònic, o bé enviar-li el program realitzat per a que el comprove el professorat. Encara que molts d'aquests cursos es creen utilitzant Moodle, aquesta plataforma no ofereix eixa interacció al moment entre el programa que està desenvolupant l'alumnat i la seua verificació.

Amb tot l'exposat anteriorment, haurem de veure de quina manera podem oferir als estudiants la possibilitat de poder continuar amb l'estudi del llenguatge de programació C/C++, davant de qualsevol imprevist o carència tecnològica. Les eines de desenvolupament on-line poden ser una bona solució. D'aquesta manera atendríem la diversitat i donaríem solucions a les diferents circumstàncies que es poden presentar a cadascun dels alumnes.

Actualment han aparegut en la xarxa cursos massius en línia (MOOC) que utilitzen entorns de desenvolupament a distància per a facilitar la pràctica en l'aprenentatge de la programació i les seues aplicacions. Això vol dir que l'estudiant disposarà d'un entorn per a escriure i executar codi en un llenguatge de programació de forma *online*. La única cosa que caldrà tenir és un navegador. Aquesta nova tecnologia ens pot ajudar a atendre la diversitat als centres de secundària a l'estudi del llenguatge de programació. D'aquesta manera no ens haurem de preocupar-nos d'utilitzar el mateix sistema operatiu que hi ha instal·lat a les aules d'informàtica ni instal·lar l'entorn de desenvolupament. A més, qualsevol alumne podrà treballar amb les pràctiques de programació des de qualsevol ordinador o dispositiu que incorpore un navegador. Simplement caldrà tenir connexió a Internet. Tot això facilitaria l'aprenentatge del llenguatge C/C++ ja que seria accessible des de qualsevol lloc. També facilitaria la tasca d'aprenentatge per aquells alumnes que estudien un llenguatge de programació amb la modalitat a distància.



El més important és que l'alumnat pugui utilitzar un entorn de programació en C/C++ de forma online sense tenir que preocupar-se per cap tipus d'instal·lació i configuració. En aquest IDE l'alumne introduirà el codi en C o C++ mitjançant un navegador i el servidor el compilarà, testejarà i enviarà al navegador de l'alumne els resultats obtinguts.

Una opció possible és utilitzar entorns de desenvolupament online en el llenguatge C/C++ que hi ha actualment en el mercat. Si fem una cerca per la xarxa podem trobar diferents IDE's que ens permeten programar en “cloud”. Per exemple, Cloud9 és un entorn de desenvolupament en el núvol que ens permet desenvolupar programes en C/C++ (també en JavaScript, HTML, XML, Java, etc.). La única cosa que ha de fer l'estudiant és registrar-se per començar a utilitzar l'IDE. Altres exemples són Ideone, Sourcelair, etc ...

Aquests entorns de desenvolupament en C/C++ online han d'oferir un entorn amigable i que s'adapte al perfil d'alumnat que l'utilitzarà. Encara que alguns dels diferents IDE al núvol que hi ha per a compilar i executar programes en C/C++ són molt intuïtius, no ofereixen les característiques adequades per a que es pugui fer un seguiment al perfil d'alumnat que hi ha a l'educació secundària. És per això que, amb aquest projecte es vol desenvolupar un entorn web afable per a que l'alumnat de batxillerat pugui editar, compilar i executar les pràctiques relacionades amb l'estudi del llenguatge de programació C/C++, així com testejar-les amb les proves del professor. D'aquesta manera facilitem la programació en qualsevol dispositiu que incorpore un navegador i des de qualsevol lloc, independentment de les circumstàncies temporals o permanents de cada alumne. Però a més, amb aquest servei “cloud” ens evitem la compra de llicències i/o instal·lació d'eines, així com el estar sempre pendent de les possibles actualitzacions.

### 1.3.- OBJECTIUS

Amb el plantejament fet als dos apartats anteriors, podem dir que l'objectiu principal d'aquest projecte final de carrera consisteix en desenvolupar una aplicació web que ofereixi un entorn simplificat de desenvolupament integrat (IDE) en el llenguatge C/C++, on els estudiants de batxillerat (o secundària en general) puguin iniciar-se en el seu estudi. La finalitat és proveir un entorn agradable a l'alumnat per a que pugui seguir correctament les pràctiques que proposa el professor independentment de les circumstàncies pròpies d'aquest (ubicació temporal o permanent de l'alumnat, sistema operatiu que utilitza, dispositiu emprat per a fer les pràctiques, etc.).

Aquest objectiu principal ens portarà a fer un estudi exhaustiu dels avenços tecnològics utilitzats en la construcció d'entorns de programació online. com ho són els frameworks de Javascript i els entorns LLVM. Aquests dos ens orientaran a fer una recerca i investigació de compiladors del llenguatge C/C++ que ens facin l'execució i correcció dels programes en un

entorn web. Com que l'alumnat provarà els seus programes en un navegador web i esperarà en aquest el resultat obtingut, s'haurà de fer un estudi de la tecnologia web més adient per aquests tipus d'IDE en el núvol.

Tot això ens indica que per aconseguir l'objectiu final del producte haurem d'anar assolint altres objectius més concrets que adquirirem a cadascuna de les fases del projecte. Aquest es descriuen a continuació:

### OBJECTIUS

1. Donar suport a la docència en l'aprenentatge del llenguatge C++ a l'assignatura “*Tecnologies de la Informació i Comunicació I i II*” que s'imparteixen a 1r i 2n de batxillerat.
2. Donar a conèixer una eina en el núvol que els ajude a practicar la programació en C++ treballant d'una forma autònoma o amb la col·laboració d'altres estudiants i/o professors.
3. Facilitar a l'alumnat un entorn web amigable per a que puguin realitzar les pràctiques en el llenguatge de programació C++.
4. Construir un entorn web d'aprenentatge del llenguatge C++ que possibiliti l'atenció a la diversitat per aquells alumnes que necessiten d'unes mesures especials durant tot el procés educatiu o de forma temporal
5. Facilitar l'aprenentatge del llenguatge C++ a tots els alumnes independentment dels recursos informàtics que tinguen disponibles.
6. Proporcionar un mitjà que facilite la pràctica de la programació en C++ a qualsevol plataforma informàtica.
7. Oferir una major interacció entre els alumnes i els alumnes amb el professorat, per a resoldre problemes de programació en C++ mitjançant un entorn web en el núvol.

Tots aquests objectius aniran assolint-se amb l'execució d'una sèrie de tasques seqüenciades a diferents fases. A cadascuna d'aquestes s'aconseguiran altres més específics que ajudaran a la consecució dels objectius genèrics definits anteriorment. A l'apartat “5.- *Fases del Projecte*” es descriuen les diferents fases en que s'ha dividit aquest project i els objectius específics que es volen assolir a cadascuna d'elles amb la realització de les tasques descrites a l'apartat descripció.

#### 1.4.- ESTRUCTURA DE LA MEMÒRIA

La present memòria descriu tot el procés que s'ha portat a terme per a l'elaboració de l'aplicació web que ens ajudarà a assolir els objectius marcats a l'apartat anterior. Els següents apartats descriuen tot l'estudi, anàlisi i accions realitzades per a poder anar aprofundint en el seu desenvolupament. Cadascun d'ells es centra en la realització d'unes tasques cohesionades que ajudaran a l'obtenció de l'aplicació a l'apartat següent.

En primer lloc caldrà fer un estudi de viabilitat. Aquest ens ajudarà a poder planificar i gestionar millor tan el temps com els recursos que utilitzarem per l'estudi i desenvolupament de l'aplicació final, sempre tenint present els objectius marcats i el resultat final que volem obtenir.

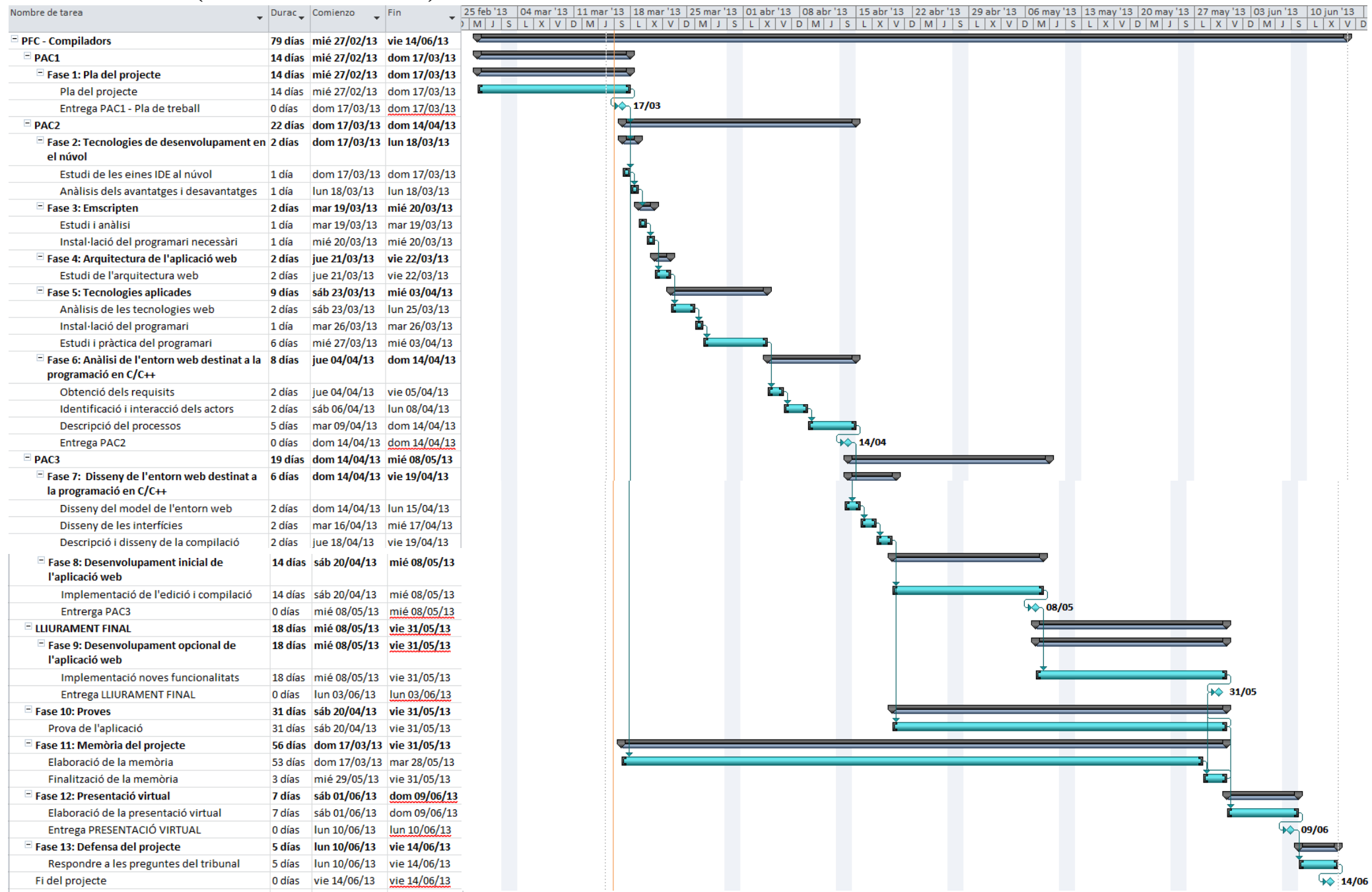
En segon lloc, es realitzarà el disseny de l'aplicació. Dins d'aquest apartat, es farà un estudi i anàlisi de les aplicacions que ja es troben disponibles a la web i que podrien ajudar als alumnes de batxillerat a poder compilar les seues pràctiques en C++ de forma *online*. A continuació veurem de quina forma podem implementar eixa compilació i execució al núvol a l'aplicació que volem construir. Tot això ens portarà a fer un anàlisi de les diferents formes que tindrem per compilar i executar una aplicació en un entorn web, la qual cosa ens ajudarà a decidir l'arquitectura més adient per al nostre producte final. Amb tota eixa informació ja estarem preparats per a elegir la tecnologia més adequada per a desenvolupar l'entorn IDE al núvol. Per últim, es farà l'estudi de l'aplicació que consistirà amb un anàlisi dels seus requisits, una descripció textual dels casos d'usos i la posterior descripció estàtica i de les interfícies que haurà de tenir l'aplicació final.

En tercer lloc, es descriurà la implementació inicial feta de l'aplicació amb les tecnologies i arquitectura elegides a l'apartat anterior. Aquesta consistirà en anar obtenint i fent allò que es va descriure a la planificació inicial. Però a més, també es veurà fins a quin punt es pot continuar amb el disseny inicial i la forma en que s'han tingut que fer modificacions com a conseqüència d'imprevistos.

Finalment, es conclourà amb les proves realitzades durant tot el procés d'elaboració del projecte i les conclusions obtingudes al final d'aquest.

# 2.- ESTUDI DE VIABILITAT

## 2.1.- PLANIFICACIÓ (DIAGRAMA DE GANTT)



## 2.2.- FASES DEL PROJECTE

## 2.2.1.- FASE 1: Pla del project

<b>PAC 1</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Elaborar el pla de treball que cal realitzar per aconseguir els objectius marcats al projecte.</li> </ul>
Descripció:	Aquest document descriu les diferents tasques i fites temporals que s'han d'executar per a obtenir el producte desitjat.
Recursos:	<ul style="list-style-type: none"> <li>• Enunciat de la proposta del PFC de Compiladors.</li> <li>• Reunió inicial amb el consultor mitjançant el servei Google Hangouts.</li> <li>• Microsoft Project.</li> <li>• Recerca d'informació a la xarxa Internet.</li> </ul>
Responsable:	L'estudiant, amb l'orientació del consultor.
Durada estimada:	14 dies (27-2-2013 al 17-3-2013)

## 2.2.2.- FASE 2: Tecnologies de desenvolupament IDE en el núvol

<b>PAC 2</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Estudiar les eines IDE que hi ha disponibles al núvol actualment.</li> <li>• Analitzar i obtenir els avantatges i desavantatges d'executar codi en el costat del client i en el costat del servidor.</li> </ul>
Descripció:	Durant aquesta tasca analitzarem com són els entorns de desenvolupament en el núvol que hi ha actualment. Aquest estudi el farem sobretot en aquells IDE que ens permeten editar, compilar i executar programes escrits en el llenguatge de programació C/C++. Exemples són

	<p>Cloud9, Ideone i Sourcelair.</p> <p>També farem un estudi de l'execució del codi 100% en un navegador i en un servidor, on extraurem els avantatges i desavantatges d'utilitzar l'un o l'altre. Aquesta fase ens servirà per a saber millor el tipus de tecnologia a utilitzar per a executar codi C/C++ al núvol.</p>
Recursos:	<ul style="list-style-type: none"> <li>• Recerca d'informació a la xarxa Internet.</li> </ul>
Responsable:	L'estudiant, amb l'orientació del consultor.
Durada estimada:	2 dies (17-3-2013 al 18-3-2013)

### 2.2.3.- FASE 3: Compilació i execució al núvol

<b>PAC 2</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Estudiar, analitzar i instal·lar tot el programari per a utilitzar el compilador Emscripten.</li> </ul>
Descripció:	<p>Uns dels avenços tecnològics que han fet possible el desenvolupament d'entorns web IDE al núvol són els <i>frameworks</i> de <i>Javascript</i> i els entorns LLVM. En primer lloc haurem de veure com compilar i executar les pràctiques realitzades pels alumnes en C/C++ en un entorn web. Per tant, estudiarem <i>emscripten</i>, un compilador de LLVM a Javascript que ens permetrà executar codi C/C++ a la web. En segon lloc, haurem de veure quina tecnologia i/o eines utilitzar per a poder construir l'entorn web del nostre IDE de C/C++ amb la informació que hem obtingut en l'estudi del compilador <i>emscripten</i>. La major part de la lògica de l'aplicació estarà relacionada amb el llenguatge Javascript i la pròpia que incorpora una aplicació web. Per últim, la instal·lació d'<i>emscripten</i> requerirà d'un programari per a la seua utilització com per exemple clang i node.js. L'estudi d'aquest últim ens aportarà una informació important de com afrontar l'aplicació web que es vol desenvolupar.</p>
Recursos:	<ul style="list-style-type: none"> <li>• Recerca d'informació a la xarxa Internet.</li> </ul>

	<ul style="list-style-type: none"> <li>• Apunts de les assignatures “<i>Compilador I</i>” i “<i>Compilador II</i>”.</li> </ul>
Responsable:	L'estudiant, amb l'orientació del consultor.
Durada estimada:	2 dies (19-3-2013 al 20-3-2013)

#### 2.2.4.- FASE 4: Arquitectura de l'aplicació Web

<b>PAC 2</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Estudiar i establir l'arquitectura web més adequada per a poder desenvolupar un entorn web on els alumnes puguin compilar, executar i comprovar les pràctiques que fan utilitzant el llenguatge C/C++.</li> </ul>
Descripció:	Tota la informació que hem obtingut a la fase anterior ens servirà per a establir l'arquitectura del programari que anem a utilitzar en la construcció d'aquest entorn. Aquesta descripció la farem per escrit i utilitzant un llenguatge descriptor d'arquitectures com UML.
Recursos:	<ul style="list-style-type: none"> <li>• Recerca d'informació a la xarxa Internet.</li> <li>• Consulta dels materials UOC de les assignatures “<i>Enginyeria del programari orientat a l'objecte</i>”, “<i>Arquitectura de sistemes distribuïts</i>” i “<i>Enginyeria del programari de components i sistemes distribuïts</i>”.</li> <li>• Programari per a modelar dades en UML.</li> </ul>
Responsable:	L'estudiant, amb l'orientació del consultor.
Durada estimada:	2 dies (21-3-2013 al 22-3-2013)

## 2.2.5.- FASE 5: Tecnologies aplicades

<b>PAC 2</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Investigar, analitzar i estudiar les tecnologies més adequades per a desenvolupar l'entorn web per a l'IDE de C/C++.</li> <li>• Instal·lar tot el programari per a poder desenvolupar l'aplicació web.</li> </ul>
Descripció:	<p>A partir de l'estudi del compilador <i>emscripten</i> i amb l'arquitectura web decidida per a implementar el producte final, obtindrem un idea de quina tecnologia de programari caldrà utilitzar per a implementar la lògica de l'aplicació web i les seues interfícies. Es decidirà els <i>frameworks</i> de <i>Javascript</i> més adequats per a desenvolupar aquesta aplicació web així com altres tecnologies que caldrà utilitzar en una comunicació web.</p> <p>Com que la infraestructura a desenvolupar és en el núvol, els alumnes utilitzaran el navegador per a connectar-se amb l'entorn IDE mitjançant la web. Per tant, en aquesta fase s'haurà d'estudiar la tecnologia a utilitzar en el costat de l'alumne i en el costat on s'ha de compilar, executar i verificar el codi C/C++ que envia l'alumnat. A més també haurem d'analitzar i elegir les eines més adequades que ens poden donar suport en la construcció de l'entorn web.</p> <p>A partir de tota eixa informació s'analitzarà i estudiarà el motor de Javascript i generador d'events node.js, el framework express.js del costat del servidor i el framework jquery del costat del client, la llibreria socket.io per a gestionar events en temps real mitjançant una connexió TCP, la gestió d'accés a l'aplicació web amb la integració del framework passport en node.js i la utilització de la base de dades documental mongodb.</p> <p>En un principi ens centrarem en les tecnologies necessàries per editar les activitats en codi C/C++ i executar-les. Per tant, potser que algunes de les esmentades al paràgraf anterior no s'utilitzen una volta finalitzat el projecte.</p>
Recursos:	<ul style="list-style-type: none"> <li>• Recerca d'informació a la xarxa Internet.</li> </ul>



	<ul style="list-style-type: none"> <li>• Programari per a desenvolupar l'aplicació web</li> </ul>
Responsable:	L'estudiant, amb l'orientació del consultor.
Durada estimada:	9 dies (23-3-2013 al 03-4-2013)

### 2.2.6.- FASE 6: Anàlisi de l'entorn web destinat a la programació en C/C++

<b>PAC 2</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Obtenir els requisits funcionals i no funcionals de l'aplicació web</li> <li>• Identificar els actors que participen en el sistema i la seua interacció amb els processos de l'aplicació web</li> <li>• Fer una descripció pas a pas d'aquests processos.</li> </ul>
Descripció:	<p>Durant aquesta fase obtindrem una descripció completa del comportament de l'aplicació web que volem desenvolupar. Aquesta la definirem en termes de requisits de programari, distingint entre els funcionals i els no funcionals. Els primers indicaran funcionalitats del programari i els descriurem utilitzant els casos d'ús, i aquests els descriurem pas a pas. Els segons estaran relacionats amb atributs de qualitat com puga ser el rendiment, la seguretat, etc...</p>
Recursos:	<ul style="list-style-type: none"> <li>• Recerca d'informació a la xarxa Internet.</li> <li>• Consulta del material UOC de l'assignatura "<i>Enginyeria del programari orientat a l'objecte</i>".</li> <li>• Programari per a modelar dades en UML.</li> </ul>
Responsable:	L'estudiant, amb l'orientació del consultor
Durada estimada:	8 dies (4-4-2013 al 14-4-2013)

## 2.2.7.- FASE 7: Disseny de l'entorn web destinat a la programació en C/C++.

<b>PAC 3</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Identificar i simbolitzar l'arquitectura, el model conceptual i de dades de l'aplicació web.</li> <li>• Dissenyar la interfície gràfica.</li> <li>• Descriure i dissenyar la compilació i execució del codi C/C++ creats pels alumnes.</li> </ul>
Descripció:	<p>Una volta hem obtingut tots els requeriments necessaris per a la nostra aplicació, obtindrem una representació gràfica d'aquests utilitzant el llenguatge de modelatge UML. Aquest models ens ajudaran a realitzar la implementació de l'entorn web utilitzant les tecnologies estudiades anteriorment.</p> <p>Per la representació de l'aplicació web mitjançant interfícies, aquestes hauran de ser intuïtives i fàcils d'utilitzar. En la construcció dels seus prototips, per una posterior implementació en una tecnologia, farem ús d'una eina lliure que ens facilite el seu disseny, com per exemple Pencil.</p>
Recursos:	<ul style="list-style-type: none"> <li>• Recerca d'informació a la xarxa Internet.</li> <li>• Consulta dels material UOC de les assignatures “<i>Enginyeria del programari orientat a l'objecte</i>”, “<i>Interfícies multimèdia</i>” i “<i>Interacció humana amb els ordinadors</i>”.</li> <li>• Programari per a modelar dades en UML.</li> </ul>
Responsable:	L'estudiant, amb la orientació del consultor
Durada estimada:	6 dies (14-4-2013 al 19-4-2013)

## 2.2.8.- FASE 8: Desenvolupament inicial de l'aplicació web.

<b>PAC 3</b>	
Objectiu:	<ul style="list-style-type: none"> <li>Implementar l'aplicació web amb la funcionalitat bàsica d'edició i compilació de codi C/C++.</li> </ul>
Descripció:	<p>En aquesta fase es començarà a construir l'aplicació utilitzant les tecnologies elegides a partir del disseny realitzat. Inicialment es començarà dotant a l'aplicació de les funcions requerides per a que els alumnes puguin editar i compilar les seues pràctiques en C/C++ a l'entorn web.</p> <p>També haurà d'incorporar un espai on el professorat pugui presentar els enunciats de les pràctiques que cal realitzar així com possibles orientacions didàctiques per al seu desenvolupament. Aquestes poden referenciar als conceptes estudiats a cada unitat de treball. Per a facilitar aquesta informació a l'alumnat, l'entorn web disposarà d'un espai de consulta per a que es pugui visualitzar els apunts i/ orientacions que el professor ha donat a classe o ha incorporat de nou per a la pràctica en qüestió.</p>
Recursos:	<ul style="list-style-type: none"> <li>Programari instal·lat a la fase 5</li> <li>Anàlisi i disseny desenvolupat de l'aplicació web a la fase 6 i 7</li> </ul>
Responsable:	L'estudiant, amb l'orientació del consultor
Durada estimada:	14 dies (20-4-2013 al 8-5-2013)

## 2.2.9.- FASE 9: Desenvolupament opcional de l'aplicació web.

<b>LLIURAMENT FINAL</b>	
Objectiu:	<ul style="list-style-type: none"> <li>Afegir noves funcionalitats a l'aplicació web</li> </ul>
Descripció:	<p>Aquesta fase estarà molt relacionada amb el treball realitzat a l'anterior. Potser que s'estiga desenvolupant noves funcionalitats a l'aplicació o, per contra, encara es continue amb les tasques de l'anterior.</p>

	<p>En un principi, durant aquesta fase se li afegirà a l'entorn web de programació funcionalitats que faciliten la realització de les pràctiques a l'alumnat: visualització jeràrquica de les pràctiques realitzades, interacció amb altres estudiants i amb el professor, consulta dels materials entregats pel professorat, avaluació de les pràctiques realitzades, ...</p>
Recursos:	<ul style="list-style-type: none"> <li>• Programari instal·lat a la fase 5</li> <li>• Anàlisis i disseny desenvolupat de l'aplicació web a la fase 6 i 7</li> </ul>
Responsable:	L'estudiant, amb l'orientació del consultor
Durada estimada:	18 dies (8-5-2013 al 31-5-2013)

## 2.2.10.- FASE 10: PROVES.

<b>PAC 2 - PAC 3 - LLIURAMENT FINAL</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Comprovar el bon funcionament de l'aplicació.</li> </ul>
Descripció:	<p>Es crearan una sèrie de proves per a verificar que els alumnes poden editar i compilar codi C/C++. Si donat el cas, se li han afegit més funcionalitats a l'aplicació web, en aquesta fase també s'inclouran aquelles proves que les validen.</p> <p>Aquesta fase estarà present durant tot el procés de desenvolupament de l'aplicació web. No sols es refereix a les proves unitàries per a comprovar que l'aplicació edita i compila codi C/C++, sinó que també inclou qualsevol altra prova per assegurar-se el bon funcionament de l'aplicació</p>
Recursos:	<ul style="list-style-type: none"> <li>• Anàlisis i disseny desenvolupat de l'aplicació web a la fase 6 i 7</li> <li>• Aplicació web desenvolupada a les fase 8 i 9</li> <li>• Conjunt de pràctiques realitzades en C/C++.</li> </ul>
Responsable:	L'estudiant, amb l'orientació del consultor
Durada estimada:	31 dies (20-4-2013 al 31-5-2013)

--	--

## 2.2.11.- FASE 11: Memòria del projecte.

<b>PAC2 – PAC3 - Lliurament final</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Elaborar la memòria del projecte.</li> </ul>
Descripció:	<p>Des de que completem la confecció del pla de treball fins que finalitzem cadascuna de les fases, anirem redactant en un document tot el que estem fent. Al finalitzar el projecte tindrem una memòria que reflectirà tot el treball realitzat per a obtenir el producte desitjat. Aquesta haurà de donar una perspectiva global de tot el que s'ha fet per a que el tribunal d'avaluació pugui formular les preguntes que considere necessàries.</p>
Recursos:	Tota la documentació elaborada i programari utilitzat durant les fases anteriors.
Responsable:	L'estudiant, amb l'orientació del consultor.
Durada estimada:	56 dies (17-3-2013 al 31-5-2013)

## 2.2.12.- FASE 12: Presentació virtual.

<b>Presentació Virtual</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Elaborar una presentació virtual</li> </ul>
Descripció:	<p>En aquesta fase es construirà un vídeo, com a molt de 20 minuts, on es sintetitzarà tot el treball realitzat i el producte obtingut</p>
Responsable:	L'estudiant, amb l'orientació del consultor
Durada estimada:	7 dies (1-6-2013 al 9-6-2013)

## 2.2.13.- FASE 13: Defensa del projecte

<b>Defensa del projecte</b>	
Objectiu:	<ul style="list-style-type: none"> <li>• Respondre a les qüestions plantejades pel tribunal d'avaluació.</li> </ul>
Descripció:	Es contestaran les preguntes que formulen els integrants del tribunal d'avaluació al fòrum de l'assignatura.
Recursos:	<ul style="list-style-type: none"> <li>• Tot el material elaborat durant el projecte.</li> </ul>
Responsable:	L'estudiant
Durada estimada:	7dies (10-6-2013 al 14-6-2013)

## 2.3.- ENTORN DE DESENVOLUPAMENT

Per a realitzar aquest projecte s'utilitzaran els següents recursos:

- Maquinari – Tot el treball es realitzarà sobre un portàtil amb les següents característiques:
  - Procesador Intel® Core™ i5 CPU M 480 @ 2.67GHz × 4
  - 4 GB de memòria RAM
- Programari – El programari utilitzat per al desenvolupament del projecte és:
  - Sistema operatiu Ubuntu 12.10 (el portàtil incorpora un sistema dual de *windows* i *linux*)
  - Paquet ofimàtic LibreOffice 3.6.2.2
  - Microsoft Project 2010.
  - Aplicació per a especificar les notacions UML a l'anàlisi i disseny de l'aplicació.
  - Aplicació per a crear els prototips de les interfícies.
  - Programari per a crear l'aplicació web: Encara que aquesta s'haurà d'estudiar detalladament abans de començar a construir el producte final, el que si sabem és que haurà d'estar relacionada amb els *frameworks* de Javascript i amb el compilador

*Emscripten* que utilitzarem per a compilar codi C/C++. A la descripció de la fase 5 (Tecnologies aplicades) s'han especificat algunes de les tecnologies que ens poden servir per al correcte desenvolupament de l'aplicació web.

En tot cas, cal dir que alguns dels programaris d'aquest llistat poden ser modificats per altres a mesura que es va treballant sobre el projecte.

- Accés a Internet.

## 2.4.- METODOLOGIA A SEGUIR

El conjunt de procediments a utilitzar per aconseguir l'objectiu principal del projecte els podem classificar en tres:

- Procediments de recerca d'informació– Aquests es refereixen a la investigació, anàlisi, estudi i proves que caldrà fer per conèixer quines són les característiques més importants del producte que volem construir. Aquesta cerca d'informació ens servirà per a poder triar la solució més viable en la construcció d'un entorn web on els alumnes puguin realitzar les pràctiques en codi C/C++.
- Procediments en la construcció del programari – Per a la construcció de l'aplicació web ens basarem en el cicle de vida del programari iteratiu i incremental estudiat a l'assignatura “*Enginyeria del programari orientat a l'objecte*”. Concretament aplicarem les quatre etapes definides en el cicle de vida de Rational Unified Process: recollida de requisits (fases 2, 3, 5), anàlisi i disseny (fases 4, 6 i 7), realització (fases 8 i 9) i prova (fase 10).
- Procediments en la planificació d'un projecte – L'enfocament metodològic que s'utilitzarà en aquest projecte es basa en la planificació orientada a objectius. Partirem d'uns objectius inicials en el treball per anar obtenint els productes desitjats. Aquest enfocament de dalt a baix es va estudiar a l'assignatura “*Metodologia i gestió de projectes informàtics*”.

## 2.5.- RESULTAT FINAL

Com a resultat del desenvolupament del projecte, obtindrem els següents productes:

- Memòria del projecte – En aquest document redactarem tota la feina que realitzarem durant les diferents fases del projecte. A part d'explicar de quina manera desenvoluparem l'entorn web, també s'especificarà l'estudi de les eines utilitzades, les decisions preses en cada moment i les possibles alteracions del pla de treball.

- Entorn web destinat a la pràctica del llenguatge C/C++ – Serà l'eina de programari resultant de tot el treball realitzat, la qual servirà per a que els alumnes puguin realitzar les pràctiques en el llenguatge de programació C/C++
- Presentació virtual – Vídeo on l'estudiant farà una presentació sintetitzada del projecte realitzat.



### 3.- DISSENY

#### 3.1.- TECNOLOGIES DE DESENVOLUPAMENT IDE EN EL NÚVOL

##### 3.1.1.- Introducció

Una volta els alumnes de l'assignatura de “*Tecnologies de la Informació i Comunicació*” de batxillerat i cicles formatius han estudiat els coneixements bàsics de la programació estructurada i practicat la construcció d'algorismes amb diagrames de flux i pseudocodi, es comença amb l'estudi del llenguatge C++. En primer lloc, caldrà preparar les estacions Lliurex de l'aula d'informàtica amb un compilador de C++ que traduísca el codi escrit a un binari que es puga executar a la màquina. En segon lloc, també s'haurà de tenir instal·lat un editor on poder escriure el codi C++. És convenient que aquest incorpore característiques visuals que ajuden a l'alumnat a identificar les estructures que s'estan utilitzant, per exemple mitjançant el ressalt de la sintaxis i el seguiment del codi escrit. I en tercer lloc, també caldrà un espai on poder compilar i executar els programes per comprovar el seu correcte funcionament.

Una de les opcions més usades és la instal·lació del paquet compilador gcc, el paquet editor Kate i el seu emulador Konsole. En canvi, altres professors opten per instal·lar, a més del compilador gcc, un entorn de desenvolupament integrat (IDE) com Anjuta per a escriure, compilar i executar les pràctiques realitzades en C++. Encara que aquestes dues opcions són propostes vàlides, que a més venen instal·lades a les distribucions Lliurex que ofereix Conselleria al centres d'educació secundària, no poden garantir una atenció diversificada a tot l'alumnat en l'estudi del llenguatge de programació C++. Per tant haurem de considerar altres eines que ens puguin ajudar a atendre eixa diversificació: estudis presencials o a distància, utilització de diferents plataformes a la de Lliurex, falta de recursos tecnològics i formació en aquests, incapacitats temporals de l'alumnat, estudi i pràctica del llenguatge C++ fora del centre educatiu, etc.. Una de les solucions que en un principi es presenta factible és la utilització de les tecnologies de desenvolupament en el núvol.

Aquest recurs està vinculat amb el concepte “*Cloud Computing*”, és a dir, oferir serveis de computació a través de la xarxa Internet. Aquest nou paradigma (“*Cloud Computing*”) ens facilitarà la utilització d'aplicacions informàtiques sense necessitat de tenir coneixements de gestió de maquinari i programari. Per tant, ens oblidarem de tenir que instal·lar, configurar i actualitzar les aplicacions que volem fer servir per a un objectiu en concret. A més, l'accés a aquestes es poden fer independentment de la plataforma i maquinari utilitzat, així com de la nostra situació geogràfica. Tan sols caldrà un dispositiu que incorpore un navegador i amb el qual puguem accedir-hi a Internet. Sembla ser que aquesta opció s'ha de tenir prou en compte en el desenvolupament de la tasca docent a l'assignatura “*Tecnologies de la Informació i Comunicació*” en un centre de

secundària, ja que possibilita poder atendre les diferents opcions o característiques que presenta l'alumnat de batxillerat i d'un cicle formatiu.

Si ens fixem amb aquest paradigma, actualment podem trobar diferents eines de desenvolupament integrat en C++ al núvol. Aquests entorns de desenvolupament integrat *on line* ens permeten editar, compilar i executar el codi font de les aplicacions que estem fent utilitzant una eina web integrada. Tot això ens facilitarà la pràctica de la programació en C++ des de qualsevol lloc, en qualsevol plataforma i sense tenir que instal·lar les eines necessàries al nostre ordinador per a poder practicar amb el llenguatge C++.

### 3.1.2.- IDE's AL NÚVOL

Alguns dels IDE de C++ que podem trobar en el núvol són:

#### SourceLair

És un IDE de programació per a diferents llenguatges, entre els quals està C++. Presenta una interfície molt agradable i intuïtiva amb diferents espais, on l'usuari pot editar, compilar i executar codi en C++, a més de desar les pràctiques en el núvol i així poder treballar amb elles des de qualsevol lloc. També permet instal·lar-lo com una extensió de Google Chrome i com una app Android. Una volta tenim escrita la pràctica en C++, el codi és enviat als servidors de SourceLair per a que el compilen i ens envien el resultat.

#### Cloud9

Aquest entorn de desenvolupament integrat en el núvol ha estat implementat en JavaScript per la companyia Cloud 9 Inc. Suporta diferents llenguatges de programació entre els quals està el C++. Aquest programari ha estat desenvolupat en `node.js`<sup>1</sup>, un entorn JavaScript del costat del servidor. Es pot accedir de dues maneres:

- De forma remota amb una inscripció prèvia. Si aquesta es fa de forma gratuïta tots els projectes es faran públics, en canvi, si es vol mantenir els projectes de forma privada i sols per aquells usuaris autenticats, caldrà subscriure's amb una quota mensual.
- Instal·lant-lo en un servidor propi i accedint-hi de forma privada. En aquest cas caldrà instal·lar abans altres requeriments, com per exemple `node.js`.

Una de les característiques més importants que ofereix aquest IDE és que podem escriure el codi de manera col·laborativa i en temps real, per tant, tots els col·laboradors que hi participen a la

---

<sup>1</sup> Node.js és un framework basat en Javascript que permet escriure aplicacions d'alt rendiment en el servidor i que actualment té un gran suport a nivell mundial. Més endavant es farà una descripció més detallada.

sessió poden provar, compilar i executar el codi amb el qual estan treballant. Però també dona suport per al mode *offline*, ja que ens permet treballar em mode local i sincronitzar al moment en que tornem a tenir connexió a Internet. Tanmateix ens permet crear el nostre propi espai de treball amb un terminal linux real on podem executar les ordres que considerem. Aquest el podem elegir com allotjament web (node, html5, php,...), com una maquina client (hosted), o com un espai de treball FTP i SSH per accedir al nostre servidor i així gestionar els projectes de forma remota. Altra característica que ofereix és integració amb Github<sup>2</sup>.

### Ideone

Compilador i depurador de codi de programació capaç de treballar amb més 40 llenguatges de programació, entre els qual es troba C++. La seua interfície també és molt simple i intuïtiva. En primer lloc s'elegeix el llenguatge que anem a utilitzar. A continuació s'escriu o es pega el codi font i les dades d'entrada a les caixes de text que hi ha a l'entorn. Per últim marquem o desmarquem executar codi privat si no volem llistar el codi a la pàgina de codi recent.

### Compileonline

Web que ens proporciona un entorn per a compilar i executar diferents tecnologies web i llenguatges de programació. Presenta una interfície fàcil d'utilitzar amb diferents espais de treball, on l'alumne pot escriure el codi font i els arguments, si cal, que hi ha que passar-li a l'executable. També ens proporciona enllaços a tutorials d'alguns dels llenguatges de programació que es podem utilitzar, com per exemple C++.

### Codepad

A més de C++, pot treballar amb altres llenguatges de programació. A part de ser un intèrpret/compilador també és una eina de col·laboració simple que ens permet compartir el codi si no el fem privat, ja que ens proporciona un enllaç per a compartir-lo.

Per a comprovar el funcionament de cadascun d'aquest IDE's al núvol, he provat els següents dos exemples de codi en C++ a tots ells, obtenint els mateixos resultats:

---

<sup>2</sup> Github és un servei d'allotjament per a projectes de desenvolupament de programari que utilitza Git com a eina de control e versions. Per tant, aquest IDE ens permetrà allotjar els projectes que desenvolupem (de forma pública o de forma provada amb pagament) en una plataforma de desenvolupament col·laboratiu com Github, a més de poder mantenir un control de versions distribuït utilitzant l'eina Git.

EXEMPLE 1	EXEMPLE 2
<pre>#include&lt;iostream&gt; using namespace std; int main() {     cout &lt;&lt; "Hola mundo";     return 0; }</pre>	<pre>#include&lt;iostream&gt; using namespace std; int main() {     int radio;     float area, perimetro;     // SALIDA: mensaje un pantalla     cout &lt;&lt; "Introduce el radio del circulo: ";     //ENTRADA: recibir dato desde teclado     cin &gt;&gt; radio;     // calculos     area = 3.1416 * radio * radio;     perimetro = 3.1416 * radio * 2;      //SALIDA: resultado en pantalla     cout &lt;&lt; "El area es " &lt;&lt; area &lt;&lt; " y el perimetro " &lt;&lt; perimetro;     cin.get();cin.get();     return 0; }</pre>

### 3.1.3.- Espais d'un IDE al núvol

A partir dels entorns que proposen els anteriors cinc IDE's utilitzats al núvol per a programar en C++, podem obtenir inicialment alguns dels espais que podria proporcionar l'eina web que volem desenvolupar. Aquests els podem classificar si són propis de l'alumnat o del professorat:

#### ESPAIS-ALUMNAT

- Visualitzar la pràctica a fer i els continguts associats a la pràctica.
- Escriure el codi font.
- Provar, compilar i executar el codi.
- Visualitzar els resultats de la compilació i de l'execució dels test unitaris.
- Gestionar les pràctiques desenvolupades.
- Estudiar i/o col·laborar amb altres companys.

- Consultar al professorat.

#### ESPais-PROFESSORAT:

- Gestionar les pràctiques que ha de fer l'alumnat.
- Escriure els test unitaris que han de comprovar per a cada pràctica
- Resolució de dubtes de les pràctiques.

#### 3.1.4.- Compilació i execució als IDE's remots

Altre aspecte a tenir en compte és la forma en que es realitza la compilació i execució als diferents IDE analitzats anteriorment. Als IDE SourceLair, Ideone, Compileonline i Codepad es realitza prement el botó corresponent, però a Cloud9 es compila al terminal virtual amb la instrucció “`gcc -o arxiu arxiu.cpp`” i s'executa amb el binari obtingut “`./arxiu`”. Per a veure la forma en que es realitza eixa compilació i execució, hauríem de comprovar les tasques que es realitzen des de que escrivim el codi en C++ fins que indiquem la seua execució per a visualitzar el resultat al navegador. Mitjançant DOM podem accedir a la jerarquia d'objecte del navegador i així poder veure els elements que conformen la pàgina web de l'IDE que estem visualitzant. L'inspector DOM ens facilita una sèrie de panells on podem comprovar els elements que conformen la pàgina. Per a veure de quina forma es realitza la compilació i execució del codi C++ anirem al panell “*Network*” per veure les crides que es realitzen. Una volta analitzat aquest panell als cinc IDE's descrits anteriorment, obtenim el següent resultat:

- Al prémer el botó per a executar el codi se li envia al servidor de l'IDE les dades introduïdes al formulari del seu entorn web via http. Aquestes són el codi C++ que ha introduït l'alumne a l'espai corresponent i altra informació que puga aparèixer en altres espais de l'entorn (per exemple el llenguatge que estem utilitzant).
- El servidor processarà la informació rebuda i retornarà el resultat obtingut via http informant de l'eixida obtinguda de la compilació i execució, o bé, dels error detectats que s'han detectat. La seua visualització es farà a l'espai de l'entorn reservat per a eixa finalitat.

Aquestes tasques es realitzen utilitzant el protocol http, el qual funciona amb la realització de peticions fetes per un client (navegador de l'alumne) i l'enviament de respostes d'un servidor (servidor de cada IDE remot). Aquesta forma de treballar ens indica que la compilació i execució del codi C++ es realitza a cadascun dels servidors, i per tant, el resultat d'aquesta operació es visualitza als navegadors del alumnes. En aquest cas, tot el codi que escriu l'alumne es prova,

compila i executa al costat del servidor.

### 3.1.5.- Conclusió

Tot l'anàlisi anterior ens dona una idea de com construir l'eina web per a que els alumnes de batxiller i cicles formatius puguin provar, compilar i executar les pràctiques realitzades en C++. En aquesta aplicació web distingirem les funcions que farà el servidor i el client. El primer haurà de rebre una petició del client per a que compile i execute un codi font en C++. Utilitzarà una eina que faci aquestes accions i que es pugui executar al costat del servidor per a obtenir el resultat obtingut. Seguidament li enviarà la resposta en forma de document HTML on es visualitzarà la solució obtinguda a la compilació i execució. D'aquesta manera es presentarà el resultat a l'usuari que va fer la petició inicial.

## 3.2.- COMPILACIÓ I EXECUCIÓ AL NÚVOL.

### 3.2.1.- Introducció

El primer que haurem de fer és veure quin compilador utilitzarem al costat del servidor per a poder enviar el resultat de provar, compilar i executar codi C++ al navegador dels alumnes. Aquest resultat s'haurà d'incloure en un document HTML com resposta a la petició que ha fet el navegador on l'alumne està realitzant les pràctiques proposades. Una opció possible és utilitzar al costat del servidor el mateix compilador que s'usa als IDE locals. En aquest cas el resultat obtingut és un binari que hauríem d'incloure al document HTML per a que s'execute al navegador del client. Per motius de seguretat aquesta acció no es pot fer, ja que un executable pot portar virus o ser un programa maliciós. A més, aquesta solució utilitzaria tan el costat del servidor com el costat del client, i això també comporta certs problemes de seguretat. Altra possibilitat és posar en marxa l'executable al propi servidor i capturar el resultat per a incloure'l al document HTML que hem d'enviar al navegador del client. Aquesta solució comporta el consum de molts recursos en el servidor i afegir-li una nova funcionalitat per a que pugui executar els binaris resultants de les pràctiques dels alumnes que estan utilitzant l'entorn web.

Uns dels propòsits d'aquest projecte és la de proporcionar a l'alumnat un entorn web d'aprenentatge del llenguatge C++ que siga independent dels recursos informàtics disponibles i de la plataforma informàtica utilitzada per accedir a aquest entorn. Aquesta filosofia és seguida per Java, un dels llenguatges més utilitzats en el desenvolupament d'aplicacions web. Aquest és independent del maquinari en que s'execute, ja que la seua compilació no consisteix en obtenir un binari, si no en traduir el codi Java a un codi intermedi que pugui ser interpretat en qualsevol dispositiu que incorpore una màquina virtual de java. Aquesta idea és la que es podria utilitzar per a realitzar les

tasques de compilació i execució de codi C++ per a que es pugui executar en un entorn web. D'aquesta manera, qualsevol aplicació desenvolupada en C++ es podria executar de forma *online* si aquesta es poguera traduir a un codi intermedi i aquest a un llenguatge més relacionat als llenguatges utilitzats a la web. L'aparició d'infraestructures pel desenvolupament de compiladors LLVM (*Low Level Virtual Machine*) han fet possible que certs llenguatges de programació, entre ells C++, es puguin executar dins d'un entorn web. A continuació, farem un anàlisi d'Emscripten, un d'aquests compiladors LLVM a Javascript.

### 3.2.2.- Emscripten

Emscripten és un programa que tradueix el codi intermedi LLVM a JavaScript. LLVM és una infraestructura de compilació que ajuda a poder executar a la web qualsevol programa fet en un llenguatge de programació que incorpore un compilador a codi intermedi LLVM. Podem dir que LLVM és com una capa d'abstracció que serveix per a que molts llenguatges de programació es puguin traduir a codi intermedi LLVM i aquest a Javascript.

Per tant, si volem executar el codi C++ al servidor, haurem de disposar d'un compilador de C++ que el tradueixi a codi intermedi LLVM. A continuació aquest serà traduït a Javascript amb emscripten, el qual haurà de ser interpretat al costat del servidor per una plataforma que done aquesta funcionalitat. El resultat s'enviarà al client incrustat dins d'un document HTML com a resposta a la petició del navegador que està utilitzant un alumne.

### 3.2.3.- Instal·lació

Per poder provar el funcionament d'Emscripten, en primer lloc caldrà instal·lar i configurar tot el programari necessari per a poder traduir codi C++ a JavaScript. Aquest muntatge s'ha realitzat a partir de la informació que ens proporciona la pàgina <https://github.com/kripken/emscripten/wiki>. A l'apartat “*Getting Started*” hi ha un enllaç a “*Getting Started on Ubuntu 12.10*”, on se'ns explica pas per pas com instal·lar Emscripten en Ubuntu 12.10. A continuació s'especifica el programari que s'ha anat instal·lant-se i una petita descripció d'aquest:

- Programari obligatori:

Nom del paquet	Descripció
<i>git</i>	Programari de control de versions distribuït.
<i>subversion</i>	Sistema de control de versions centralitzat.

<i>cmake</i>	Generador de compilació multiplataforma
<i>build-essential</i>	Metapaquet (paquet de paquets que s'encarreguen d'instal·lar altres paquets al que fa referència) que conté les instruccions per a instal·lar els paquets essencials per a programar en C++.
<i>default-jre</i>	JRE és el conjunt de programes i llibreries de java per a executar aplicacions desenvolupades en aquest llenguatge de programació.

- Programari front-end del llenguatge C++ (Clang) a LLVM
  - S'obté i s'instal·la LLVM i Clang (3.2) des del repositori centralitzat amb el programari subversion. Clang s'haurà instal·lat en `~/llvm32build/bin`.
  - A continuació es configura Clang en el path per a que es pugui executar des de qualsevol lloc de la màquina local.
- Node.js
  - Es realitza la instal·lació i configuració de node.js (al mateix temps s'instal·la Python 2.7.3) seguint les passes que s'indiquen a l'adreça url <https://github.com/joyent/node/wiki/Installing-Node.js-via-package-manager>
  - Node.js és un entorn de Javascript del costat del servidor que utilitza un model asíncron i orientat a esdeveniments. Usa el motor de JavaScript V8 de Google, una màquina virtual molt ràpida i de gran qualitat que Google incorpora al seu navegador Chrome. Per tant podem dir que node és un sistema de tractament d'esdeveniments d'entrada i eixida en JavaScript (s'espera un esdeveniment i s'escriu una funció de devolució d'eixe esdeveniment). Amb node també podem construir aplicacions web. Per fer una similitud amb el servidor web Apache, podem dir que Node és un Apache sense cap mòdul inclòs.
  - Al mateix temps també s'ha instal·lat npm (*Node Package Manager*) que ens facilitarà la compilació, instal·lació i actualització de mòduls així com les seues dependències. Serà el programa que controlarà els mòduls en node.js.
- Codi Emscripten
  - S'obté des de Github, una plataforma de desenvolupament col·lectiu de programari per allotjar projectes utilitzant el sistema de versions Git



### 3.2.4.- Comprovació

Una volta ja tenim tot el programari instal·lat i configurat, passarem a fer la primera prova per veure com es compila i executa un arxiu cpp a node.js i com s'obté una resposta en format JavaScript. A la carpeta d'emscripten hi ha una carpeta anomenada test que incorpora programes en codi C++. Anem a fer la prova amb el típic “*Hola mon*”:

- La següent figura ens mostra el codi C++ que emscripten compilarà i l'arxiu JavaScript obtingut com a resultat d'eixa compilació

```
joan@joan-Satellite-L670:~/emscripten$ cat ./tests/hello_world.cpp
#include<stdio.h>

class Test {}; // This will fail in C mode

int main() {
    printf("hello, world!\n");
    return 1;
}

joan@joan-Satellite-L670:~/emscripten$ ./em++ ./tests/hello_world.cpp
joan@joan-Satellite-L670:~/emscripten$ ls *js
a.out.js
joan@joan-Satellite-L670:~/emscripten$
```

- A continuació executarem l'arxiu *a.out.js* a node.js

```
joan@joan-Satellite-L670:~/emscripten$ node a.out.js
hello, world!
joan@joan-Satellite-L670:~/emscripten$
```

- Ara posarem una errada en la compilació. El missatge d'aquest error ve del compilador Clang que no ha pogut generar el codi LLVM correctament.

```
joan@joan-Satellite-L670:~/emscripten$ ./em++ ./tests/hello_world.cpp
./tests/hello_world.cpp:6:28: error: expected ';' after expression
    printf("hello, world!\n")
                           ^
1 error generated.
emcc: compiler frontend failed to generate LLVM bitcode, halting
joan@joan-Satellite-L670:~/emscripten$
```

### 3.2.5.- Compilació - Execució

Anteriorment hem conclòs que tota la part de prova, compilació i execució es realitza al costat del servidor. Amb la combinació dels compiladors Clang (C++ a LLVM) i Emscripten (LLVM a JavaScript) obtenim la traducció a Javascript de qualsevol aplicació escrita en C++. D'aquesta manera podem executar qualsevol aplicació escrita en C++ dins d'un entorn web. Les diferents formes que tenim per a poder realitzar aquestes tasques en un entorn web són:

#### CLIENT-SERVIDOR (EXECUCIÓ AL COSTAT DEL SERVIDOR)

- Al costat del servidor s'interpreta el codi Javascript generat pel compilador emscripten utilitzant una plataforma que ho permeta com node. El resultat obtingut

es captura i s'envia al navegador de l'alumne inserit a la pàgina web. En aquest cas el codi s'executa al 100% al costat del servidor. Les avantatges i desavantatges que es deriven d'aquesta arquitectura són:

### **Avantatges**

- x L'execució de les aplicacions en el servidor es realitza sense tenir en compte el tipus de client que li ha fet eixa petició. Aquesta execució es realitzarà en un ambient controlat, i la resposta s'enviarà en un format normalitzat per a que qualsevol tipus de client pugui interpretar-la.
- x Crear o instal·lar una aplicació que proveï, compili i executi codi C++ en la part del servidor és més segura ja que s'utilitza el xifrat de la informació i es protegeix la informació davant qualsevol intent d'intrusió.
- x Els servidors tenen un major nombre de recursos que afavoreixen la prova, compilació i execució de les aplicacions escrites en C++ que poden necessitar d'un maquinari mínim per ser executades (per exemple aplicacions gràfiques en 3D).
- x S'allibera de càrrega de treball als clients. Això permet que els alumnes puguin accedir-hi al servei amb qualsevol tipus de dispositiu que els permeti connectar-se a Internet, independentment dels recursos de maquinari i programari que tinga

### **Desavantatges**

- x Cal implementar l'aplicació al servidor que realitzi tota la lògica de negoci de l'entorn web.
- x Caldrà implementar un sistema de seguretat per a mantenir la integritat de les dades que hi ha al servidor i per a que sols puguin accedir-hi aquells alumnes que s'han enregistrat prèviament.
- x També s'haurà d'incloure un sistema que detecte codi maliciós a les peticions rebudes pels alumnes. Potser que un alumne programe un virus que impedisca el funcionament de l'entorn web.
- x Es pot sobrecarregar el servidor ja que haurà de provar, compilar i executar el codi C++ de totes les peticions rebudes pels alumnes.

- ✗ Davant una fallida inesperada en el servidor l'aplicació quedaria suspesa per a tots els clients.
- ✗ L'alumne ha de tenir sempre una connexió permanent per a poder utilitzar l'entorn web.

#### CLIENT-SERVIDOR (EXECUCIÓ AL COSTAT DEL SERVIDOR I DEL CLIENT)

- Al costat del servidor es compila el codi C++ i s'obté un arxiu Javascript. Aquest s'inclou a la pàgina web que s'enviarà al navegador de l'alumne per a que interprete aquest codi. En aquest cas la compilació es realitza al costat del servidor mentre que l'execució (interpretació del Javascript resultant) al costat del client. Aquesta última acció requerirà que el navegador de l'alumne tinga habilitada la funcionalitat d'interpretar Javascript. Actualment tots els navegadors la incorporen. Aquesta forma de realitzar les tasques de compilació i execució tan al costat del servidor com al costat del client, porta implícit els avantatges i desavantatges d'executar el codi tan al costat del servidor com al costat del client. Els avantatges i desavantatges d'executar el codi Javascript al costat del client són:

##### **Avantatges**

- ✗ S'allibera la càrrega d'execució al servidor, però no la de compilació.
- ✗ No cal una aplicació servidora que interprete llenguatge Javascript.

##### **Desavantatges**

- ✗ Cal tenir habilitada la funció d'interpretació de Javascript al navegador.
- ✗ La interpretació del Javascript utilitza recursos al costat del client.
- ✗ La seguretat, ja que pot contenir codi maliciós que s'executarà al costat del client.
- ✗ L'alumne pot visualitzar les proves que el professor ha utilitzat.
- ✗ Com s'executa el codi al costat del client, no es pot fer cap tipus de registre de les pràctiques que realitza cada alumne al costat del servidor.

#### CLIENT-CLIENT (EXECUCIÓ AL COSTAT DEL CLIENT)

- L'altra forma de treballar consisteix en realitzar les tasques de compilació i execució al costat del client. Quan l'alumne sol·licita fer la prova, compilació i execució del

codi C++, el servidor rebrà aquesta sol·licitud i generarà una pàgina web que incorpore el codi escrit per l'alumne, les proves a realitzar i el compilador que traduirà tota la pràctica de C++ a Javascript de la mateixa manera que ho fa emscripten, però executant-se en el navegador. En aquest cas el 100% de l'execució es farà al costat del client.

Un exemple de compilador-intèrpret que permet treballar d'aquesta manera és mruby, el qual compila i executa codi Ruby<sup>3</sup> al costat del client. A l'anàlisi fet sobre les tecnologies IDE al núvol no vaig trobat cap IDE que operara d'aquesta manera. Però amb la recerca d'informació del compilador emscripten i l'ajuda del consultor he comprovat a l'adreça <http://qiezi.me/projects/mruby-web-irb/mruby.html#> el funcionament d'aquest intèrpret. La inspecció dels elements DOM al panell Network no dona cap activitat, indicant que la compilació i execució del codi Ruby que hi ha a l'editor es realitza al mateix navegador, sense necessitat de cap *plugins*. Aquesta forma de treballar comporta les següents avantatges i desavantatges:

### Avantatges

- ✗ S'allibera de càrrega al servidor, tan de la compilació com de l'execució. Actuarà com un servidor web que respondrà les sol·licituds amb una pàgina que continga el codi C++ a provar, compilar i executar, i el compilador-intèrpret que haurà de compilar-lo i executar-lo al mateix navegador.
- ✗ No cal una aplicació servidor que interprete llenguatge Javascript.
- ✗ No cal tenir cap *plugin* al navegador del client.

### Desavantatges

- ✗ La compilació i execució del codi Ruby amb mruby al costat del client utilitza recursos d'aquest, per tant els recursos de la màquina on s'executa el navegador hauran de ser suficients per a poder compilar i executar el codi C++.
- ✗ La seguretat, ja que pot contenir codi maliciós que s'executarà al costat del client.

---

<sup>3</sup> Ruby és un llenguatge de programació interpretatiu, reflexiu i orientat a objectes, creat pel japonés Yukihiro “Matz” Matsumoto. Combina una sintaxis inspirada en Python i Perl amb característiques de programació orientat a objectes similars a Smalltalk.

- × L'alumne pot visualitzar les proves que el professor ha utilitzat.
- × Com s'executa el codi al costat del client, no es pot fer cap tipus d'enregistrament o seguiment de les pràctiques que realitza cada alumne al costat del servidor.
- × S'ha d'incloure el compilador-intèrpret a la pàgina web que s'envia a l'alumne juntament amb el codi a provar, compilar i executar. Per tant es poden enviar com a resposta pàgines de certa grandària que necessiten més recursos de memòria al costat del client i que també pot repercutir en el temps de resposta.

Altra de les tecnologies que treballa amb el codi C++ al navegador és *Client Native*. Aquesta tecnologia de codi obert permet executar codi C i C++ dins d'un navegador, mantenint la seguretat de tecnologies web com JavaScript. La seua arquitectura és la d'un compilador i entorn de seguretat. El codi C++ es descarrega al navegador on, una volta se li han aplicat les proves de seguretat per a que no afecte al sistema de l'equip de l'usuari, es compila i s'executa. La seua forma d'actuar és diferent a l'exposades anteriorment. No es basa amb la utilització d'una màquina virtual si no que s'aprofita de la potència que tenen actualment les màquines dels usuaris per a executar-se en aquesta.

### 3.2.6.- Conclusions

Emscripten és un compilador de codi intermedi LLVM a JavaScript que ens possibilita poder executar aplicacions C++ a la web. JavaScript és un llenguatge interpretat que s'utilitza principalment en la part del client on el navegador s'encarrega d'executar-lo. Però amb node.js aquest es pot interpretar al costat del servidor ja que incorpora el motor V8 JavaScript. Per tant, davant una petició web per a compilar i executar una aplicació C++, aquesta es compilarà amb clang per a obtenir un arxiu escrit en codi intermedi LLVM, el qual es traduirà a Javascript utilitzant emscripten. A continuació node.js l'executarà i presentarà la resposta a la petició inicial amb l'enviament d'una pàgina HTML que contindrà el resultat de l'execució de l'arxiu traduït a JavaScript. Aquesta forma de funcionar representa una arquitectura client – servidor on el client utilitza el navegador com a interfície i el servidor com a eina de compilació i execució.

Altra possibilitat que s'ens ha presentat és de treballar com una arquitectura client – servidor, però a diferència de l'anterior, les tasques de compilació i execució es troben repartides als dos costats. Per una part, el servidor s'utilitzarà com a eina de compilació i el navegador de client com a interfície que a més executarà (interpretarà) el codi traduït a JavaScript.

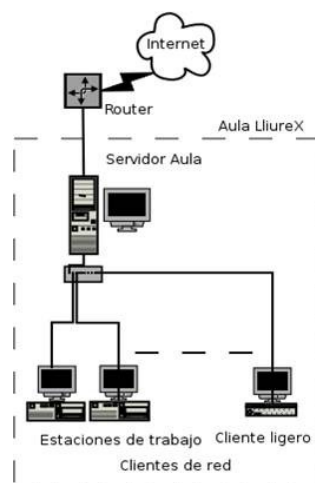
Per últim, també s'ha vist una arquitectura client – client en la que l'eina de compilació i execució es trobe situada al navegador de l'alumne amb la finalitat de compilar i executar les pràctiques dels alumnes al costat del client. L'interpret mruby, com a compilador de C a JavaScript utilitzant Emscripten, ens facilita tenir l'eina de compilació i execució al navegador sense necessitat d'utilitzar cap plugin.

Encara que aquest apartat estava dedicat al compilador Emscripten i les seues possibilitats per a desenvolupar un entorn web on els alumnes puguen treballar amb el codi C++, s'ha considerat mencionar la tecnologia Native Client com una eina en la qual es pot compilar i executar codi C++ al navegador de l'alumne per a executar les aplicacions als seus ordinadors. Aquesta forma de treballar també entra dins de l'arquitectura client – client però amb la diferència que per a la seua execució s'aprofita dels recursos del sistema de l'alumne, la qual cosa està fora dels objectius establerts per a desenvolupar aquesta eina.

### 3.3.- ARQUITECTURA DE L'APLICACIO WEB

#### 3.3.1.- Introducció

Abans de començar a especificar el tipus d'arquitectura més adient per a l'entorn web que volem desenvolupar, cal tenir present el model d'aula utilitzat a les aules d'informàtica dels centres de secundària. Aquest model d'aula d'informàtica desenvolupat en Lliurex consisteix en una xarxa formada per un servidor al qual es poden connectar tan estacions de treball (client d'aula) com clients lleugers. El servidor disposa de dues targetes de xarxa. La primera li serveix per a treballar dins de l'aula d'informàtica atenen les peticions que li arriben dels equips clients i la segona li permet connectar-lo a l'exterior (Internet o altra xarxa del centre). El següent gràfic representa aquest model:



Els equips clients poden funcionar de diferents modes:

- Client de xarxa – El PC arranca amb el sistema operatiu instal·lat al seu disc dur però utilitza el servidor per a obtenir les dades de configuració (adreça IP, dades d'usuari, DNS, etc ..)
- Client lleuger – El PC arranca un sistema mínim i inicia una sessió gràfica al servidor. Bàsicament és un terminal gràfic i els programes s'executen al servidor.
- Estació de treball – El PC arranca amb el sistema operatiu local i no utilitza la xarxa ni el servidor per a obtenir les seues dades de configuració.

Un dels objectius d'aquest model (a banda d'aquells que ajuden al professorat en l'administració i configuració de l'aula) és reutilitzar el maquinari antic que hi ha al centres. D'aquesta forma es poden mantenir operatius equips antics que pels seus requisits de maquinari no poden ser utilitzats de forma independentment amb les aplicacions actuals. A més, els nous equipaments TIC que està enviant Conselleria als centres de secundària estan formats per un servidor i diferents clients lleugers. Aquests dispositius no tenen disc dur, i per tant sistema operatiu, però que suporten una arrancada en xarxa per a iniciar una sessió gràfica en el servidor.

També hem de tenir en compte el maquinari que utilitzarà l'alumnat per connectar-se a l'eina web per a desenvolupar les pràctiques en C++. Les seues característiques poden ser molt diverses i mai podem suposar uns recursos de maquinari mínims per a poder utilitzar l'entorn IDE per a programar en C++. Per exemple, les biblioteques dels centres de secundària i municipals ofereixen clients lleugers per a que els alumnes puguin connectar-se a Internet i per a treballar amb aplicacions informàtiques des d'un servidor.

Altres dels serveis que es posa a disposició dels centres de secundària és la plataforma virtual “mestre a casa”. Aquest portal proporciona a la comunitat educativa la informació, eines i recursos necessaris per a portar a terme la tasca docent. En aquesta infraestructura tecnològica es pot crear comunitats educatives així com allotjar les pàgines webs dels centres . També ens proporciona espais virtuals per a impartir formació a distància als diferents nivells educatius.

Tot el que hem comentat anteriorment ens pot donar una idea de l'arquitectura a utilitzar en la construcció de l'aplicació web. Però a més, també ens pot indicar de quina forma treballarà aquesta aplicació, és a dir, on es realitzarà la compilació i execució del codi C++ dins d'aquest entorn web. Aquestes poden ser les següents:

- Una arquitectura web basada en el model client – servidor. El model d'aula Lliurex es basa amb una tecnologia client – servidor, en la que el servidor de l'aula proporciona diferents serveis als clients que es connecten a ell. L'eina web a desenvolupar també haurà de proporcionar aquestes funcionalitats.

- La tecnologia a utilitzar en el desplegament de l'aplicació web serà client – servidor, on el servidor s'utilitzarà com a eina de compilació i execució. La utilització de clients lleugers al model d'aula Lliurex implica l'execució de qualsevol programa al servidor de l'aula on es connecten. Si les tasques de compilació i execució es realitza al costat del client (en aquest cas el servidor de l'aula) pot implicar la saturació o la parada d'aquest, ja que el costat del client per al clients lleugers és el servidor de l'aula. Hem de tenir en compte que les característiques de maquinari del servidor de l'aula d'informàtica són molt inferiors a les que poguera tenir el servidor utilitzat per a compilar i executar les pràctiques dels alumnes. “*Mestre a casa*” pot ser la plataforma on ubicar el servidor que ha de realitzar les tasques de compilació i execució.

### 3.3.2.- Funcionament de l'IDE

Amb tot l'anàlisi fet als apartats anteriors podem veure que el funcionament bàsic de l'eina web que volem desenvolupar consisteix en l'execució dels següents passos:

- L'alumne escriu o visualitza una pràctica escrita en C++ al seu navegador.
- Sol·licita que aquesta siga compilada i executada premen un botó al navegador.
- Aquesta petició és rebuda juntament amb el codi en C++ escrit per l'usuari.
- Si l'alumne ha elegit testejar la seua pràctica se li afegirà el codi de prova del professorat a eixa pràctica en altre cas no. A continuació es realitza la compilació de la pràctica resultant.
- Seguidament es realitza l'execució d'eixa compilació.
- El resultat es visualitzarà al navegador de l'alumne.

Tota aquesta seqüenciació de tasques bàsiques es realitzaran en màquines situades a diferents llocs geogràfics ja que es tracta d'un entorn de desenvolupament en C++ al núvol. A més, aquestes treballaran amb diferents plataformes. Per tant serà una aplicació web distribuïda on els alumnes realitzaran peticions de compilació i execució de la seua pràctica en C++ en un servidor i rebran els resultats obtinguts per a cadascuna d'elles. La construcció d'aquesta eina la realitzarem basant-nos amb l'arquitectura client – servidor.

### 3.3.3.- Arquitectura client – servidor

En una arquitectura client – servidor, una aplicació es modela com un conjunt de components servidors, que ofereixen uns serveis i un conjunt de clients que utilitzen aquests serveis.



En aquest cas l'alumne interaccionarà amb l'aplicació web a través del navegador. Com a conseqüència de l'activitat que farà l'usuari a l'aplicació web que volem desenvolupar (compilar i executar pràctiques, emmagatzemar treballs, parlar amb altres usuaris, etc.), s'enviaran peticions al servidor, lloc on s'allotja l'aplicació que gestionarà l'activitat de l'usuari i on s'emmagatzem les dades utilitzades per aquesta. El servidor processa la petició i retorna la resposta al navegador per a que se la presente a l'usuari. En aquesta arquitectura podem diferenciar els següents elements:

- Navegadors – És la interfície de l'usuari
- Aplicació – Programari encarregat de realitzar les operacions necessàries per a efectuar les activitats que l'entorn web posa a disposició de l'usuari.
- Base de dades – On s'emmagatzema la informació relacionada amb l'aplicació (registre d'usuaris, enunciats de pràctiques, proves de cada pràctica, etc ...)

Aquesta distribució d'elements ens indica que l'arquitectura client-servidor a utilitzar es basarà en un model o arquitectura de tres capes. Per tant, l'element navegador sols servirà per a presentar la informació i no haurà de realitzar cap tipus de processament. Tota la lògica de negoci es realitzarà al servidor de l'entorn web. D'aquesta manera podem dividir l'aplicació web en tres nivells:

- Nivell de presentació – S'encarrega de generar les interfícies d'usuari en funció de les accions que realitzi aquest.
- Nivell de negoci – Conté la lògica que modela el processos de negoci i es on es realitza tot el processament necessari per a poder atendre les peticions dels usuaris.
- Nivell d'administració de dades – S'encarrega de fer persistent tot la informació. Subministra i emmagatzema tota la informació per al nivell de negoci

En aquesta aplicació web distribuïda el client sols mantindrà la presentació i la resta d'aplicació s'executarà remotament en un servidor. Els dos nivell anteriors i part del tercer es trobaran en un servidor, mentre que la resta del tercer nivell es situaria en una base dades (que pot estar situat en el mateix servidor on es realitza la lògica de negoci). En la seua forma més simple, aquesta aplicació web distribuïda vendria definida per la següent figura:



Per a desenvolupar aquesta aplicació web basada en una arquitectura client – servidor de tres nivells, on l'execució es realitza al 100% al costat del servidor, farem ús d'alguns dels patrons de disseny que ens poden ajudar en el disseny d'aquest IDE al núvol. Un dels patrons que ha demostrat ser fonamental en el disseny de les aplicacions web és el patró MVC (*Model View Controller*).

### 3.3.4.- Patró Model-Vista-Controlador

Aquest és un patró o model d'abstracció de desenvolupament de programari que separa les dades de l'aplicació, la interfície de l'usuari i la lògica de negoci en tres components diferents:

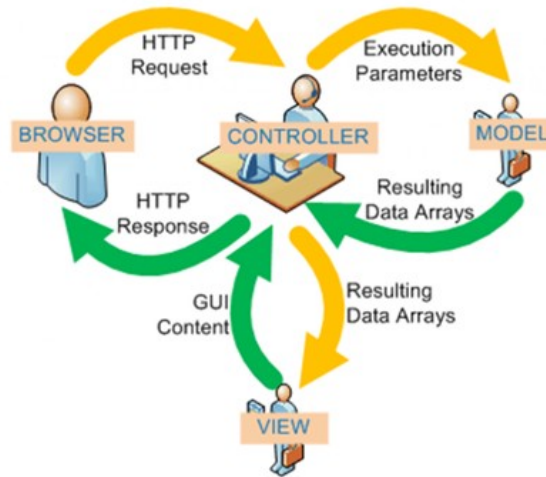
- El model – Conjunt de classes que representarà la informació que l'entorn web processarà per a que els alumnes provaran, compilaran i executaran les pràctiques en C++.
- Les vistes – Conjunt de classes que s'encarrega de visualitzar als alumnes el contingut del model on es processen les peticions que li realitzen a l'aplicació IDE al núvol.
- El controlador – Objecte que s'encarregarà de dirigir el flux de control de l'aplicació com a conseqüència de missatges externs. A partir d'aquests, el controlador s'encarregarà de modificar el model i/o obrir i tancar vistes. Per tant, el controlador té accés al model i a les vistes, però les vistes i el model no coneixen l'existència d'aquest.

El significat d'aquest patró de disseny a l'aplicació web que volem desenvolupar és:

- Una vista representa l'estat del model en un moment donat (per exemple la primera pantalla que es presenta al navegador de l'alumne). Les vistes seran pàgines HTML que l'usuari visualitzarà al seu navegador.
- Quan un controlador rep un esdeveniment disparat per l'alumne a través de la interfície (per exemple clic al botó per a executar el codi C++), accedeix al model per a executar l'acció adequada i es presenta en una nova vista el resultat d'aquesta acció. A vegades aquest controlador també pot cridar directament a una vista. En aquest cas l'usuari, mitjançant les vistes interactua amb l'aplicació enviant esdeveniments al servidor a través de peticions HTTP. Al servidor es troba el codi de control per aquests esdeveniments. En funció de l'esdeveniment concret actuarà sobre el model convenient.
- El model consistirà en el conjunt d'objectes que modelen els processos de negoci que es realitzen mitjançant el sistema (per exemple les pràctiques, els usuaris, les proves, etc ..).

- Els resultats de l'acció es tornaran a l'usuari en forma de pàgina HTML mitjançant resposta HTTP.

Per tant, el flux que segueix el control generalment és el següent:



Aquesta estratègia per a desenvolupar el projecte final ens aporta una sèrie d'avantatges i desavantatges. Tenir clarament separada la interfície, la lògica de negoci i la presentació ens ajudarà a separ el model (les classes que representaran la informació de l'IDE al núvol) de les vistes. D'aquesta manera funcionaran de forma independent, cosa que ens facilita el manteniment davant de les fallades i un possible escalament de l'aplicació en un futur. Respecte al contres, podem dir que la distribució en components obliga a crear i mantenir un major nombre d'arxius i que haurem d'elegir una tecnologia de desenvolupament que s'adeqüe a aquest paradigma. Actualment existeixen molts frameworks de desenvolupament d'aplicacions web basats en aquest patró MVC que ens ajuden en la seua construcció. Aquests frameworks web són un conjunt de components (classes, descriptors XML, arxius de configuració, etc ..) que es poden personalitzar i configurar per a construir l'IDE al núvol. Tot això ens ajudarà a reutilitzar el que ja hi ha fet i per tant agilitzarà el desenvolupament de l'entorn web.

### 3.4.- TECNOLOGIES APLICADES

#### 3.4.1.- Introducció

Una volta hem decidit l'arquitectura de l'aplicació web a desenvolupar haurem de veure quina o quines tecnologies utilitzarem per a desenvolupar l'IDE al núvol. Recordem que el disseny d'aquesta aplicació web es basa en una arquitectura client – servidor, on l'execució es realitza al costat del servidor. Per tant haurem de veure quina tecnologia usarem al costat del client i quina al

costat del servidor. A més, el patró de disseny MVC ens ajudarà en la construcció d'aquesta aplicació. Com que l'execució es realitza al costat del servidor, el model, la vista i el controlador també s'executaran en aquest costat.

### 3.4.2.- Tecnologies al costat del servidor

A continuació es descriuen les tecnologies que s'utilitzaran al costat del servidor, les quals s'encarregaran de la major part de la lògica de l'entorn web que volem desenvolupar. En aquesta elecció s'han tingut en compte els anàlisis fets als apartats anteriors.

#### **NODE.JS**

Segons els seus creadors, Node.js és una plataforma construïda sobre el motor d'execució JavaScript V8 de Google per a crear aplicacions de xarxa d'una manera fàcil i escalable. Utilitza un model d'entrada/eixida sense bloquejos basats en esdeveniments, característica que el fa lleuger i eficient, perfecte per aplicacions en temps real que tenen un gran consum de dades i que s'executen en dispositius distribuïts. A l'apartat on s'ha estudiat el compilador Emscripten hem comprovat com node.js ens possibilita executar aplicacions escrites en JavaScript al costat del servidor.

A l'aplicació que volem desenvolupar, node.js serà el servidor on s'ha d'implementar i executar tota la lògica de negoci cada volta que un alumne realitzi una petició al seu navegador mitjançant un protocol de sol·licitud/resposta (HTTP). Això implica saber quines parts de l'aplicació web cal implementar per a que un alumne es connecte a l'entorn, escriga o carregi una pràctica en C++, i la pugi provar, compilar i executar per a obtenir al seu navegador el resultat de qualsevol d'eixes accions. Bàsicament, aquestes poden ser les següents:

- Implementar un servidor HTTP per a poder servir pàgines web.
- Atendre les diferents peticions per a enrutar-les, és a dir, crear les rutes per a cada petició HTTP que li arriba.
- Associar a cada petició una funció que s'encarregue de realitzar la tasca associada a aquest esdeveniment, és a dir, associar a cada petició un handler d'esdeveniments que es faci càrrec de processar la petició. Aquesta part manipularà les peticions GET provinents de les peticions que fan els alumnes al seus navegadors i crearà gestors (*handlers*) que les despatxaran als mètodes corresponents per a produir l'eixida i retornar-la al navegador de l'usuari de forma no bloquejant.
- A l'igual que l'anterior, altra part haurà de gestionar les peticions POST de forma no

bloquejant (asíncrona). La informació que arribe del client web davant d'una petició, s'ha de processar (per exemple executar una pràctica) i a continuació s'haurà de respondre amb el resultat obtingut.

- Altra part serà la lògica de les vistes que els gestors (*handlers*) de peticions hauran d'utilitzar per a enviar el contingut apropiat al navegador de l'usuari.

Cadascuna d'aquestes part bàsiques caldrà implementar-les a la plataforma node.js. Aquestes es podrien fer des de node.js o bé utilitzar mòduls contributius que ajuden a crear els serveis que ha d'oferir el servidor en aquesta aplicació web. Una forma d'instal·lar-los fàcilment és utilitzant l'NPM (*Node Package Manager*). A continuació veurem els paquets que haurem d'afegir a node.js per a proporcionar-li tota la funcionalitat requerida per la construcció de l'IDE al núvol. La instal·lació d'aquests mòduls es poden fer locals al projecte que estem desenvolupant o global per a que es puguin utilitzar en qualsevol projecte que realitzem a node.js.

## MÒDULS

npm	NPM és un gestor de paquets per a node.js que s'utilitza per a instal·lar mòduls als projectes que estem desenvolupant. Això vol dir que quan es descarrega un mòdul, aquest s'agrega a un projecte local el qual el tindrà disponible per a poder-lo utilitzar. També ens ajuda a poder cercar actualitzacions dels paquets o mòduls que tenim instal·lats així com revisar les seues dependències per a que no hi haja conflictes entre aquests.
express	Express és un framework MVC que ens permet construir aplicacions web d'una forma ràpida, senzilla i de forma estructurada. Aquest mòdul i les seues dependències les instal·larem utilitzant npm. Ens permet crear l'estructura bàsica de l'aplicació web així com tot allò que està relacionat amb l'accés al servidor (rutes, gestió de les peticions i vistes, etc..). Aquest converteix a node.js en un servidor web que pot gestionar peticions HTTP però que al mateix temps també exerceix com a sistema MVC.
jade	Jade és un motor de plantilles que permet generar pàgines HTML. Amb aquest mòdul es generaran les vistes incloent-hi dades que es generen dinàmicament.
socket.io	Socket.io és una llibreria que ens permet gestionar esdeveniments en temps real mitjançant una connexió TCP utilitzant JavaScript. Permet la comunicació en temps real entre el servidor i el client mitjançant websockets (tecnologia que

	permet a les aplicacions web mantenir una comunicació bidireccional amb processos en el costat del servidor). Aquest mòdul ens servirà per a la comunicació en xarxa i s'executa tan en el servidor amb node.js com en el client.
passport	Passport és un middleware (programari que actua com a intermediari entre aplicacions o components de programari i de maquinari) que ens ajudarà a gestionar l'autenticació dels usuaris que utilitzen l'entorn web. Treballa amb la combinació d'Express. Per a poder fer una autenticació d'usuaris caldrà tenir una base de dades.
mysql	Mysql és un paquet que ens permet connectar-nos amb node.js a una base de dades SQL. Ens pot ajudar a gestionar els usuaris que poden connectar-se a l'aplicació.
nodemon	Aquesta eina ens ajudarà a monitoritzar el codi de l'aplicació web per a que quan aquesta canvie es reinicie node.js. D'aquesta manera ens des preocupem de tenir que reiniciar node cada volta que modifiquen el codi.

### 3.4.3.- Tecnologies al costat del client

Aquestes tecnologies s'utilitzaran en el navegador de l'usuari, per tant es tractaran de llibreries que treballaran en JavaScript i en HTML. Els processament dels efectes i funcionalitats que tinga la pàgina que l'usuari vol visualitzar al seu navegador recaurà en aquest. Com que l'accés a aquest IDE al núvol es farà des de diferents tipus d'ordinadors i/o terminals amb diferents prestacions (recordem la utilització de clients lleugers a les aules Lliurex), cal que no sobrecarregui massa al navegador de l'usuari. La tecnologia empleada serà JQuery.

JQuery	JQuery és un framework que de Javascript que ens permet simplificar la manera d'interactuar amb els documents HTML, manipular l'arbre DOM, gestionar esdeveniments, etc.. aquesta tecnologia s'utilitzarà per a desenvolupar la lògica de les interfícies d'usuari. Per tant, ens ofereix una infraestructura que ens ajudarà per a crear aplicacions al costat del client. En aquest cas l'utilitzarem per a crear les interfícies de l'usuari
--------	---

### 3.5.- ANÀLISI DE L'ENTORN WEB DESTINAT A LA PROGRAMACIÓ C++

#### 3.5.1.- Introducció

Abans de començar a utilitzar les tecnologies descrites a l'apartat anterior, cal tenir una descripció detallada del sistema a desenvolupar. Aquesta es pot especificar textualment i/o utilitzar un llenguatge gràfic que ens permeti visualitzar, especificar, construir i documentar l'aplicació. Aquesta descripció es farà identificant els requisits del sistema, els qual descriuen comportaments, propietats i restriccions del programari que hem de desenvolupar. Aquests els podem classificar en dos tipus: els funcionals i els no funcionals. Els primers especifiquen el comportament intern del programari, mentre que els segons representen restriccions que assegurin el bon funcionament de l'aplicació web. Abans de veure quins són aquests requisits seria convenient especificar genèricament l'algorisme que detalla les accions més importants que es realitzen des de un usuari es connecta a l'aplicació fins que finalitza la seua sessió.

1. El professor emmagatzema l'enunciat d'una pràctica a la base de dades.
2. El professor emmagatzema la pràctica solucionada a la base de dades.
3. El professor emmagatzema les proves d'eixa pràctica a la base de dades.
4. El professor emmagatzema el resultat de cadascuna de les proves d'eixa pràctica.
5. L'alumne desenvolupa la pràctica.
6. L'alumne executa la pràctica. Si dona errors de compilació torna al punt 5.
7. L'alumne comprova els tests d'eixa pràctica.
8. Comprova el resultat obtingut amb el que dona el professor. Si no és el mateix torna al punt 5.
9. El professor avalua la pràctica de l'alumne.

#### 3.5.2.- Requisits funcionals.

A l'aplicació web identifiquem dos actors que interaccionaran amb ella: l'alumnat i el professorat. Cadascun d'ells interaccionen amb l'aplicació realitzant activitats associades als seus rols. El primer que farem és veure quines activitats realitzaran els actors alumne i professor que correspondran amb els seus requisits funcionals.

## Professor – Requisits funcionals

- Iniciar sessió: permet al professor identificar-se per a entrar al sistema.
- Tancar sessió: permet al professor tancar i eixir del sistema.
- Crear grup: permet al professor crear un grup d'alumnes.
- Crear alumne: permet al professor crear un alumne i afegir-lo a un grup.
- Eliminar grup: permet al professor eliminar un grup d'alumnes.
- Eliminar alumne: permet al professor eliminar un alumne d'un grup.
- Crear professor: permet al professor que fa d'administrador crear un professor i assignar-li un grup.
- Llistar grups: permet al professor llistar els grup que té.
- Llistar usuaris d'un grup: permet al professor visualitzar els usuaris que hi ha en un grup.
- Llistar pràctiques: permet al professor visualitzar les pràctiques que ha de fer un grup.
- Modificar pràctica: permet al professor modificar el contingut d'una pràctica.
- Crear pràctica: permet al professor crear l'enunciat d'una pràctica.
- Afegir pràctica/grup: permet al professor afegir una pràctica a un grup o grups.
- Extraure pràctica/grup: permet al professor trencar la relació d'una pràctica amb un grup o grups.
- Eliminar pràctica : permet al professor eliminar una pràctica a un grup.
- Afegir solució: permet al professor afegir la solució en C++ d'una pràctica.
- Eliminar solució: permet al professor eliminar la solució d'una pràctica.



- Modificar solució: permet al professor modificar la solució d'una pràctica.
- Publicar solució: permet al professor fer pública la solució als alumnes.
- Visualitzar solució: permet al professor visualitzar la solució que dona a una pràctica.
- Afegir prova: permet al professor afegir la prova que es farà a una pràctica.
- Eliminar prova: permet al professor eliminar la prova que es farà a una pràctica.
- Modificar prova: permet al professor modificar la prova que es farà a una pràctica.

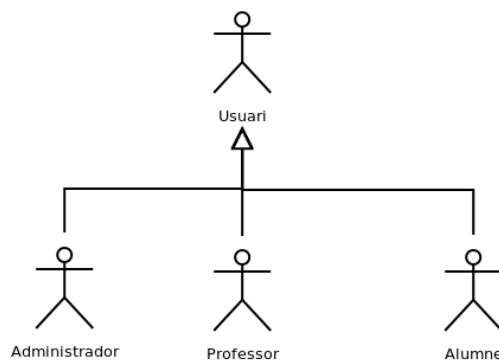
#### Alumne – Requisits funcionals

- Iniciar sessió: permet a l'alumne identificar-se per a entrar al sistema.
- Tancar sessió: permet a l'alumne tancar i eixir del sistema.
- Llistar usuaris d'un grup: permet a l'alumne visualitzar els companys del seu grup.
- Llistar pràctiques: permet a l'alumne visualitzar les pràctiques que ha de fer.
- Visualitzar pràctica: permet a l'alumne seleccionar una pràctica i veure el seu enunciat per començar a treballar.
- Crear treball: permet a l'alumne crear un nou treball per a desenvolupar el codi C++ associat a la pràctica que ha proposat el professor. Aquest s'haurà de desar si es vol tenir disponible per a pròximes sessions.
- Visualitzar treballs: permet a l'alumne veure els treballs que ha fet.
- Obrir treball: permet a l'alumne obrir per a modificar una pràctica que ja tenia feta.
- Executar treball: permet a l'alumne compilar i executar el seu treball utilitzant la seua funció main ().
- Comprovar prova: permet a l'alumne executar el seu treball utilitzant la prova

que el professor proposa al seu main ()).

- Visualitzar resultat: permet a l'alumne veure el resultat a obtenir quan se li passe la prova al codi que ha desenvolupat en C++.
- Visualitzar solució: permet a l'alumne visualitzar la solució que dona el professor a la pràctica.

Molts d'aquest requisits funcionals corresponents a activitats que es podrien agrupar com a requisits funcionals d'un tercer actor o rol que es relaciona amb l'aplicació web: l'administrador. Aquest es podria dedicar a fer les funcions pròpies de manteniment de grups i usuaris. També podem veure que hi ha altres requisits funcionals que són comuns al actors professor i alumne: per exemple iniciar sessió i tancar sessió. Però a més, també serien activitats que l'administrador hauria de realitzar per a poder entrar i eixir del sistema. Amb la incorporació d'aquest tercer actor haurem de generalitzar-los com a usuari genèric. Tot i que les funcions d'administrador en un centre de secundària sol realitzar-les un professor, caldrà considerar-lo com un usuari diferent a aquest per a que les gestions d'administració puguin ser realitzades des de fora. D'aquesta manera la gestió d'administració podrà venir d'una aplicació externa. Al següent diagrama es poden veure els actors que s'ha identificat a l'IDE al núvol:



## Actors – Casos d'ús

L'aparició d'aquests nous actors implica una nova distribució dels requisits funcionals entre ells. La definició d'aquests requisits la farem mitjançant els casos d'usos d'acord a la informació que hi ha a la següent taula:

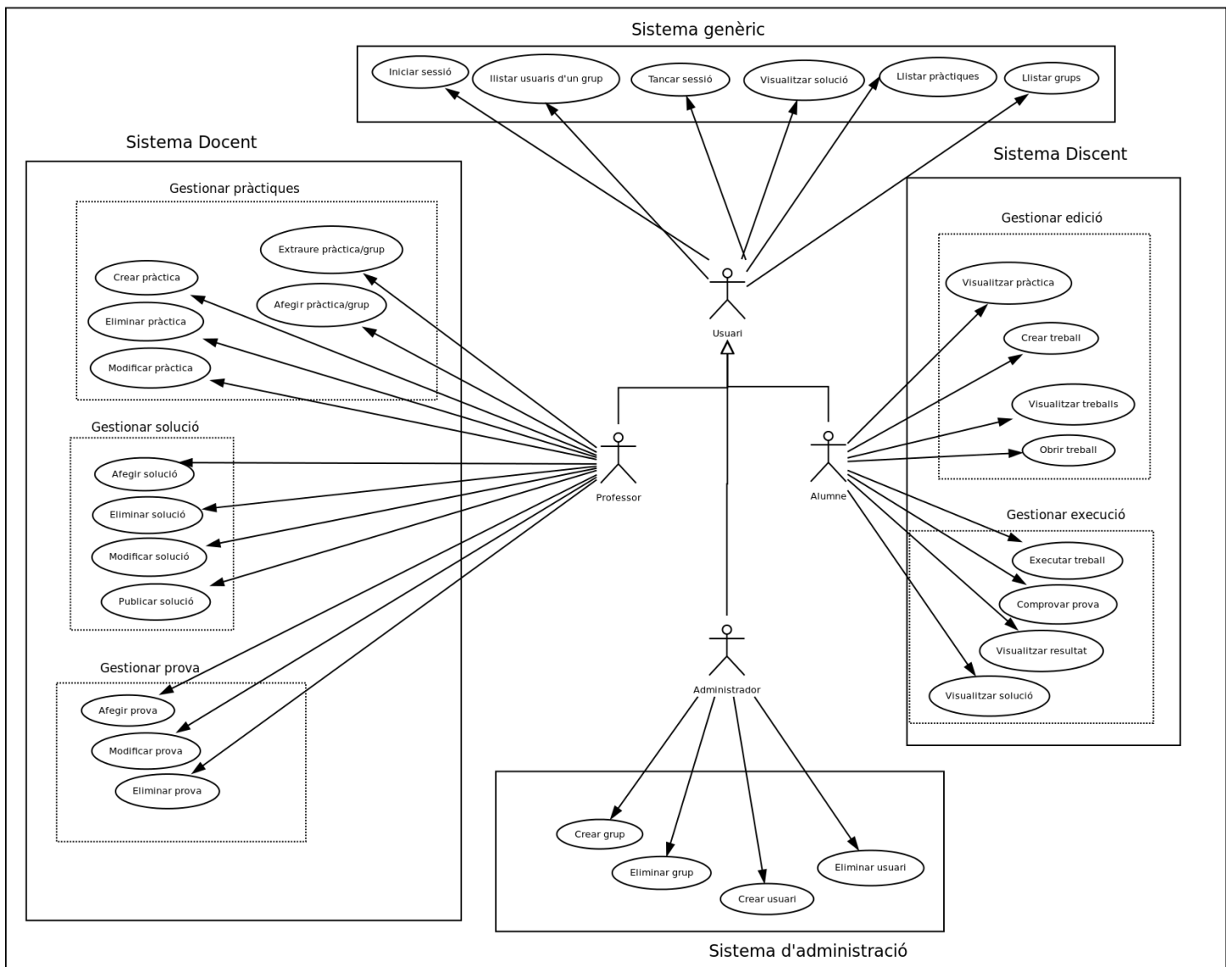
<b>Actor</b>	<b>Casos d'ús</b>
Usuari	[Iniciar sessió] - [Tancar sessió] [Llistar grups] - [Llistar usuaris d'un grup] [Llistar pràctiques] – [Visualitzar solució]
Professor	[Crear pràctica] – [Eliminar pràctica] – [Modificar pràctica] [Afegir pràctica/grup] – [Extraure pràctica/grup] [Afegir solució] - [Eliminar solució] - [Modificar solució] – [Publicar solució] [Afegir prova] - [Eliminar prova] - [Modificar prova]
Alumne	[Visualitzar pràctica] [Crear treball] – [Visualitzar treballs] – [Obrir treball] [Executar treball] – [Comprovar prova] – [Visualitzar resultat] [Visualitzar solució]
Administrador	[Crea grup] – [Eliminar grup] [Crear usuari] - [Eliminar usuari]

### 3.5.3.- Descripció dels casos d'ús.

En aquest apartat anem a descriure cadascun dels caso d'ús que té associat cada actor que intervén a l'aplicació web. Aquesta la farem textualment a partir d'una plantilla on s'especificarà tota la informació referent a cada cas d'ús, i gràficament utilitzant la notació UML. En primer lloc descriurem amb una notació UML els actors que intervenen en l'aplicació i els casos d'ús que tenen associats:

## DESCRIPCIÓ GRÀFICA

### IDE al núvol



En segon lloc anem a fer una descripció textual de cadascun d'aquests casos d'ús d'acord a la classificació feta al diagrama UML.

## DESCRIPCIÓ TEXTUAL

- **Sistema genèric**

Casos d'ús que són utilitzat per qualsevol usuari, independentment del seu rol al sistema.

## CAS D'ÚS: INICIAR SESSIÓ

Funcionalitat:	<ol style="list-style-type: none"> <li>1. L'usuari introdueix al navegador l'adreça de l'aplicació web.</li> <li>2. Es visualitza la pantalla inicial. Aquesta és un formulari amb dos camps per a introduir: nom d'usuari i contrasenya.</li> <li>3. L'usuari introdueix les seues dades i prem al botó “Iniciar sessió”.</li> <li>4. El sistema comprova que aquest usuari està enregistrat.</li> <li>5. Si les dades són incorrectes es visualitzarà un missatge d'error: “Usuari incorrecte, torna a provar-ho”. Si falta qualsevol camp del formulari per emplenar es visualitzarà altres missatge d'error: “Per favor, no pot deixar cap camp buit”.</li> <li>6. Si les dades són correctes, apareixerà al navegador de l'usuari la pantalla principal de l'aplicació. Aquesta dependrà del rol que tinga l'usuari que ha entrat al sistema.</li> </ol>
Actors:	Usuari
Precondició:	L'usuari ha de tenir un nom d'usuari i una contrasenya per a poder entrar al sistema.
Postcondició:	Un usuari ha entrat al sistema amb un rol determinat

## CAS D'ÚS: TANCAR SESSIÓ

Funcionalitat:	<ol style="list-style-type: none"> <li>1. Es prem al botó “Tancar sessió”.</li> <li>2. Els sistema prem el control i allibera qualsevol recurs que estiguera utilitzant-se.</li> <li>3. Per últim, el sistema presenta al navegador de l'usuari la</li> </ol>
----------------	---

	pantalla inicial on es demana el nom d'usuari i la contrasenya
Actors:	Usuari
Precondició:	1. L'usuari ja havia entrat anteriorment al sistema
Postcondició:	L'usuari ha abandonat l'aplicació web.

## CAS D'ÚS: LLISTAR USUARIS D'UN GRUP

Funcionalitat:	<ol style="list-style-type: none"> <li>1. Aquesta acció s'activa quan l'usuari fa un clic damunt del botó "Llistar grup".</li> <li>2. Es llisten els noms dels usuaris que formen part del grup i el seu rol (noms dels alumnes i del professor)</li> </ol>
Actors:	Usuari
Precondició:	1. Haver iniciat sessió.
Postcondició:	Una consulta no genera cap canvi al sistema.

## CAS D'ÚS: LLISTAR PRÀCTIQUES

Funcionalitat:	<ol style="list-style-type: none"> <li>1. Aquesta acció s'activa quan l'usuari fa un clic damunt del botó "Llistar pràctiques".</li> <li>2. Es llisten els noms de les pràctiques d'un grup.</li> </ol>
Actors:	Usuari
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió.</li> <li>2. Per a un professor haver seleccionar grup.</li> </ol>
Postcondició:	Una consulta no genera cap canvi al sistema.

## CAS D'ÚS: VISUALITZAR SOLUCIÓ

Funcionalitat:	<ol style="list-style-type: none"> <li>1. Aquesta acció s'activa quan l'usuari fa un clic damunt del botó “Visualitzar solució”.</li> <li>2. Es mostra el codi C++ que el professor proposa com a solució a la pràctica.</li> </ol>
Actors:	Usuari
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió.</li> <li>2. Estar visualitzant una pràctica.</li> </ol>
Postcondició:	Una consulta no genera cap canvi al sistema.

## CAS D'ÚS: LLISTAR GRUPS

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Llistar grups”</li> <li>2. Es visualitzaran tots els grups que hi ha a la base de dades.</li> </ol>
Actors:	Usuari
Precondició:	<p>Haver iniciat sessió.</p> <p>Per a un alumne o professor es llistaran els grups que tenen assignats mentres que per l'administrador es visualitzaran tots els grups.</p>
Postcondició:	Una consulta no genera cap canvi al sistema.

- **Sistema d'administració**

Casos d'ús que són utilitzat per el rol administrador. Aquest serà un usuari que també tindrà el rol de professor al sistema.

## CAS D'ÚS: CREAR GRUP

Funcionalitat:	1. S'activa en prémer el botó “Crear grup”
----------------	--

	<ol style="list-style-type: none"> <li>2. Apareix una finestra on s'indican les dades del grup (nom i descripció).</li> <li>3. Si el grup ja existeix s'informarà amb un missatge: “Grup no vàlid, ja existeix”. Confirmem i tornem al formulari per a crear el grup.</li> <li>4. Si el grup no existeix s'indica l'èxit de l'operació: “Grup <i>nom</i> creat correctament”. Confirmem</li> <li>5. Tornem a la pantalla anterior.</li> </ol>
Actors:	Administrador
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com administrador.</li> <li>2. El grup no ha d'existir a la base de dades.</li> </ol>
Postcondició:	Es dona d'alta la informació del grup a la base de dades.

## CAS D'ÚS: ELIMINAR GRUP

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Eliminar grup”</li> <li>2. Activem el cas d'ús [Llistar Grups] per fer apareix en una finestra els grup que hi ha.</li> <li>3. Es selecciona el grup. Confirmem.</li> <li>4. Activem el cas d'ús [Llistar usuaris d'un grup] i visualitzem el alumnes i professor del grup. Indiquem amb un missatge que esborraran de la base de dades.</li> <li>5. Ho confirmem. Si no es confirma es torna a la pantalla anterior.</li> <li>6. Esborrem de la base de dades els alumnes. Esborrem el professor si no té cap altre grup assignat.</li> <li>7. S'informa amb un missatge: “Grup nom esborrat amb els seus alumnes i professor”. Confirmem el missatge.</li> </ol>
----------------	--



	8. Tornem a la pantalla anterior.
Actors:	Administrador
Precondició:	1. Haver iniciat sessió com administrador. 2. El grup ha d'existir a la base de dades.
Postcondició:	S'esborra de la base de dades el grup i els alumnes. S'esborra el professor si aquest ja no té cap grup assignat.

## CAS D'ÚS: CREAR USUARI

Funcionalitat:	1. S'activa en prémer el botó “Crear usuari” 2. S'elegeix el tipus d'usuari a enregistrar: Alumne o Professor. 3. Apareix una finestra amb un formulari on s'ompliran les dades de l'usuari. 4. Es selecciona el grup on estarà assignat. S'ha d'activar el cas d'ús [Llistar grups] per a que apareguen els grups. 5. Si l'usuari existeix ho indiquem, omplint de nou el formulari. 6. Confirmem 7. Tornem a la pantalla anterior.
Actors:	Administrador
Precondició:	1. Haver iniciat sessió com administrador. 2. L'usuari no ha d'existir a la base de dades. 3. Ha d'estar donat alta el grup on volem que estiga l'alumne
Postcondició:	Es dona d'alta un usuari. En aquest cas potser un alumne o un professor.

## CAS D'ÚS: ELIMINAR USUARI

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Eliminar usuari”</li> <li>2. Apareix una finestra on indiquem el nom de l'usuari.</li> <li>3. Si no existeix ho indiquem, en altre cas visualitzem les dades de l'usuari.</li> <li>4. S'indica el missatge: “Voleu esborrar aquest usuari (s/n)?”</li> <li>5. Confirmem.</li> <li>6. Tornem a la pantalla anterior.</li> </ol>
Actors:	Administrador
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com administrador.</li> <li>2. L'usuari ha d'existir</li> </ol>
Postcondició:	S'esborra un alumne o professor de la base de dades.

- **Sistema docent**

## Gestionar pràctica

## CAS D'ÚS: CREAR PRÀCTICA

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Crear pràctica”</li> <li>2. S'ompli el formulari amb les dades de la pràctica: títol, enunciat, etc...</li> <li>3. Es confirma l'operació.</li> <li>4. Si l'operació és correcta s'indica a la pantalla amb un missatge. En altre cas hi haurà un missatge d'error.</li> </ol>
Actors:	Professor

Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a professor.</li> <li>2. No ha d'existir una pràctica amb el mateix identificador.</li> </ol>
Postcondició:	Es desa a la base de dades la pràctica.

## CAS D'ÚS: AFEGIR PRÀCTICA/GRUP

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Afegir pràctica/grup”</li> <li>2. Es visualitzaran les pràctiques que ha creat el professor.</li> <li>3. Es selecciona una o varies pràctiques.</li> <li>4. Es visualitzaran els grups que té assignats el professor que ha iniciat sessió.</li> <li>5. Es seleccionaran un o més grups on es realitzarà la pràctica.</li> <li>6. Es confirma la operació.</li> <li>7. Si l'operació és correcta s'indica a la pantalla amb un missatge. En altre cas hi haurà un missatge d'error.</li> </ol>
Actors:	Professor
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a professor.</li> <li>2. El professor haurà creat les pràctiques que vol assignar.</li> <li>3. Ha d'existir els grups on volem assignar la pràctica o pràctiques.</li> <li>4. Les pràctiques no han d'estar relacionades amb els grups seleccionats.</li> </ol>
Postcondició:	Es desa a la base de dades les relacions pràctica – grup

## CAS D'ÚS: EXTRAURE PRÀCTICA/GRUP

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Extraure pràctica/grup”</li> <li>2. Es visualitzaran les pràctiques que ha creat el professor.</li> </ol>
----------------	---

	<ol style="list-style-type: none"> <li>3. Es selecciona una o varies pràctiques.</li> <li>4. Es visualitzaran les pràctiques seleccionades amb els grups on estan afegides</li> <li>5. Es selecciona la relació pràctica - grup que es vol trencar (extraure eixa pràctica d'eixe grup.)</li> <li>6. Es confirma la operació.</li> <li>7. Si l'operació és correcta s'indica a la pantalla amb un missatge. En altre cas hi haurà un missatge d'error.</li> </ol>
Actors:	Professor
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a professor.</li> <li>2. Les pràctiques a extraure d'un grup han d'haver estat abans afegides a eixe grup.</li> </ol>
Postcondició:	S'esborra de la base de dades les relacions seleccionades entre pràctica i grup.

## CAS D'ÚS: ELIMINAR PRÀCTICA

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Eliminar pràctica”</li> <li>2. Es visualitzaran les pràctiques que ha creat el professor que ha iniciat sessió.</li> <li>3. Es selecciona la pràctica o pràctiques a esborrar.</li> <li>4. Es visualitzaran el nom o noms de les pràctiques a eliminar i els grups on estaven afegides (si ho estaven).</li> <li>5. Es confirma l'acció.</li> </ol>
Actors:	Professor
Precondició:	1. Haver iniciat sessió com a professor.

	2. Ha d'existir la pràctica o pràctiques.
Postcondició:	S'esborra de la la base de dades la o les pràctiques seleccionades i les seues relacions amb els grups si les tenien.

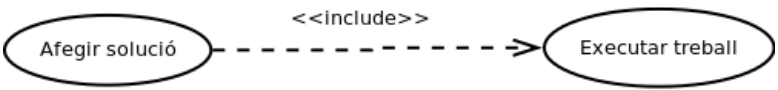
## CAS D'ÚS: MODIFICAR PRÀCTICA

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Modificar pràctica”</li> <li>2. Es visualitzen les pràctiques que ha creat el professor que ha iniciat sessió</li> <li>3. Es selecciona la pràctica a modificar.</li> <li>4. Es visualitza la informació referent a la pràctica.</li> <li>5. Es modifica la informació pertinent.</li> <li>6. Es confirma la modificació.</li> <li>7. Si l'operació és correcta s'indica a la pantalla amb un missatge: “Modificació realitzada”. En altre cas hi haurà un missatge d'error.</li> </ol>
Actors:	Professor
Precondició:	<p>Haver iniciat sessió com a professor.</p> <p>Ha d'existir la pràctica</p>
Postcondició:	Es sobreescriu la pràctica seleccionada a la base de dades.

## Gestionar solució

## CAS D'ÚS: AFEGIR SOLUCIÓ

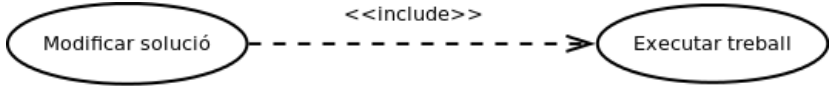
Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Afegir solució”</li> <li>2. Es visualitzaran les pràctiques que ha creat el professor que ha</li> </ol>
----------------	---

	<p>iniciat sessió.</p> <p>3. Selecció de la pràctica.</p> <p>4. Si no té solució assignada escriurem la solució o la carregarem des d'un arxiu. En altre cas ens indicarà que ja en té una i per tant confirmem i tornem a la pantalla principal.</p> <p>5. Comprovem els correcte funcionament. En aquest cas caldrà poder-la executar amb el cas d'ús [Executar treball] del rol alumne. Això implica que [Afegir Solució] fa ús de [Executar Treball].</p>  <pre> graph LR     A([Afegir solució]) -.-&gt; &lt;&lt;include&gt;&gt;  B([Executar treball])     </pre> <p>6. Confirmem aquest arxiu com a solució oficial d'eixa pràctica per a eixe grup.</p> <p>7. Ocultem la solució als alumnes.</p>
Actors:	Professor
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a professor.</li> <li>2. Ha d'existir la practica.</li> <li>3. No ha d'existir una solució a la pràctica.</li> </ol>
Postcondició:	Es desa a la base de dades la solució associada a eixa pràctica.

## CAS D'ÚS: ELIMINAR SOLUCIÓ

Funcionalitat:	<ol style="list-style-type: none"><li>1. S'activa en prémer el botó “Eliminar solució”</li><li>2. Es visualitzaran les pràctiques que ha creat el professor que ha iniciat sessió.</li><li>3. Si té solució assignada, es visualitzarà. En altre cas s'informa de l'error.</li><li>4. Confirmem l'eliminació.</li></ol>
Actors:	Professor
Precondició:	<ol style="list-style-type: none"><li>1. Haver iniciat sessió com a professor.</li><li>2. Ha d'existir la practica.</li><li>3. Ha d'existir una solució a la pràctica.</li></ol>
Postcondició:	S'elimina de la base de dades la solució associada a una pràctica.

CAS D'ÚS: MODIFICAR SOLUCIÓ

<p>Funcionalitat:</p>	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Modificar solució”</li> <li>2. Es visualitzaran les pràctiques que ha creat el professor que ha iniciat sessió.</li> <li>3. Selecció de la pràctica.</li> <li>4. Si té solució assignada es visualitzarà, en altre cas s'informarà de l'error.</li> <li>5. Modifiquem la solució o importem una de nou.</li> <li>6. Comprovem el correcte funcionament.</li> </ol>  <pre> graph LR     A([Modificar solució]) -.-&gt; &lt;&lt;include&gt;&gt;  B([Executar treball])     </pre> <ol style="list-style-type: none"> <li>7. Confirmem la modificació.</li> </ol>
<p>Actors:</p>	<p>Professor</p>
<p>Precondició:</p>	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a professor.</li> <li>2. Ha d'existir la practica.</li> <li>3. Ha d'existir una solució a la pràctica.</li> </ol>
<p>Postcondició:</p>	<p>Es modifica a la base de dades la solució que està associada a eixa pràctica</p>

CAS D'ÚS: PUBLICAR SOLUCIÓ

<p>Funcionalitat:</p>	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Publicar solució”</li> <li>2. Es visualitzaran les pràctiques que ha creat el professor que ha iniciat sessió.</li> <li>3. Selecció de la pràctica.</li> <li>4. Si té solució assignada es visualitzarà, en altre cas s'informarà</li> </ol>
-----------------------	--

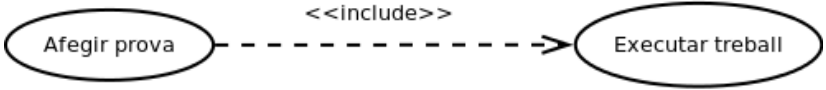


	<p>de l'error.</p> <p>5. Publiquem la solució per a que els alumnes la puguem vore.</p> <p>6. Confirmem la publicació.</p>
Actors:	Professor
Precondició:	<p>1. Haver iniciat sessió com a professor.</p> <p>2. Ha d'existir la practica.</p> <p>3. Ha d'existir una solució a la pràctica.</p> <p>4. La solució ha d'estar com a privada</p>
Postcondició:	Es modifica a la base de dades l'accés a la solució associada a la pràctica seleccionada

### Gestionar prova

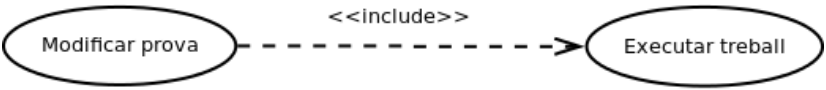
#### CAS D'ÚS: AFEGIR PROVA

Funcionalitat:	<p>1. S'activa en prémer el botó “Afegir prova”</p> <p>2. Es visualitzaran les pràctiques que ha creat el professor que ha iniciat sessió.</p> <p>3. Es selecciona la pràctica.</p> <p>4. Si encara no té solució assignada s'informarà amb un missatge: “Primer has d'assignar una solució”.</p> <p>5. Si té una solució assignada. S'escriu o s'importa la prova a fer.</p> <p>6. S'uneixen la solució i la prova en un arxiu, Comprovem el funcionament. Per tant:</p>
----------------	---

	 <p>7. El resultat el desarem. Serà el que l'alumne comprovarà quan prove el seu codi si els resultat és l'esperat o no.</p>
Actors:	Professor
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a professor.</li> <li>2. Ha d'existir la practica.</li> <li>3. Ha d'existir una solució a la pràctica.</li> </ol>
Postcondició:	<p>Es desa a la base de dades la prova feta a la solució d'una pràctica.</p> <p>Es desa a la base de dades el resultat obtingut en eixa prova feta a la solució d'una pràctica</p>

CAS D'ÚS: MODIFICAR PROVA

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Modificar prova”</li> <li>2. Es visualitzaran les pràctiques que ha creat el professor que ha iniciat sessió.</li> <li>3. Es selecciona la pràctica.</li> <li>4. Si encara no té solució assignada s'informarà amb un missatge: “Primer has d'assignar una solució”.</li> <li>5. Si té una solució assignada, es visualitzaran les proves que hi ha per a eixa solució.</li> <li>6. Es selecciona una prova.</li> <li>7. Es modifica o s'importa la modificació.</li> <li>8. S'uneixen la solució i la modificació en un arxiu, Comprovem el funcionament. Per tant:</li> </ol>
----------------	---

	 <p>9. El resultat obtingut serà sobreescrit sobre el resultat que tenia la prova anterior.</p>
Actors:	Professor
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a professor.</li> <li>2. Ha d'existir la practica.</li> <li>3. Ha d'existir una solució a la pràctica.</li> <li>4. Ha d'existir la prova per a eixa solució.</li> </ol>
Postcondició:	<p>Es sobreescriu a la base de dades la prova modificada feta a la solució d'una pràctica.</p> <p>Es sobreescriu a la base de dades el nou resultat obtingut en eixa prova feta a la solució d'una pràctica.</p>

CAS D'ÚS: ELIMINAR PROVA

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Eliminar prova”</li> <li>2. Es visualitzaran les pràctiques que ha creat el professor que ha iniciat sessió.</li> <li>3. Es selecciona la pràctica.</li> <li>4. Si encara no té solució assignada s'informarà amb un missatge: “Primer has d'assignar una solució”.</li> <li>5. Si té una solució assignada es visualitzaran les proves que se li han fet.</li> <li>6. Seleccionem la prova a eliminar.</li> <li>7. Confirmen l'acció.</li> </ol>
Actors:	Professor

Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a professor.</li> <li>2. Ha d'existir la practica.</li> <li>3. Ha d'existir una solució a la pràctica.</li> <li>4. Aquesta solució ha de tenir proves.</li> </ol>
Postcondició:	S'esborra a la base de dades la prova feta a la solució d'una pràctica i el resultat associat a eixa prova.

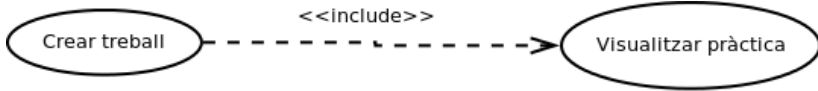
○ **Sistema discent**

### Gestionar edició

#### CAS D'ÚS: VISUALITZAR PRÀCTICA

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Visualitzar pràctica”</li> <li>2. Apareixeran les pràctiques que ha de realitzar en el seu grup..</li> <li>3. Selecciona un pràctica.</li> <li>4. Es visualitza l'enunciat.</li> </ol>
Actors:	Alumne
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a alumne.</li> <li>2. Ha d'existir la pràctica.</li> </ol>
Postcondició:	La consulta de l'enunciat d'una pràctica no genera cap canvi al sistema.

CAS D'ÚS: CREAM TREBALL

<p>Funcionalitat:</p>	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Crear treball”</li> <li>2. S'escriu codi C++ per a resoldre l'enunciat de la pràctica o s'importa d'una altre arxiu.</li> <li>3. Es va desant cada cert temps</li> </ol> <p>Per a crear treball abans s'ha tingut que visualitzar la pràctica, per tant:</p>  <pre> graph LR     A([Crear treball]) -.-&gt; &lt;&lt;include&gt;&gt;  B([Visualitzar pràctica])             </pre>
<p>Actors:</p>	<p>Alumne</p>
<p>Precondició:</p>	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a alumne.</li> <li>2. Ha d'existir la pràctica.</li> <li>3. S'esta veien l'enunciat de la pràctica.</li> </ol>
<p>Postcondició:</p>	<p>Es desa a la base de dades el treball que està realitzant un alumne d'un grup per a una pràctica.</p>

CAS D'ÚS: VISUALITZAR TREBALLS

<p>Funcionalitat:</p>	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Visualitzar treballs”</li> <li>2. Es visualitzen els treballs que ha fet l'alumne.</li> </ol>
<p>Actors:</p>	<p>Alumne</p>
<p>Precondició:</p>	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a alumne.</li> <li>2. Han d'existir treballs</li> </ol>
<p>Postcondició:</p>	<p>No genera cap canvi al sistema.</p>

CAS D'ÚS: OBRIR TREBALL

<p>Funcionalitat:</p>	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Obrir treballs”</li> <li>2. Es visualitzen els treballs que ha fet l'alumne, per tant:             <div style="text-align: center; margin: 10px 0;"> <pre>                     usecaseDiagram                         ObrirTreball --&gt; VisualitzarTreballs : &lt;&lt;include&gt;&gt;                     </pre> </div> </li> <li>3. Es selecciona el treball.</li> <li>4. Es visualitza l'enunciat de la pràctica d'eixe treball, per tant:             <div style="text-align: center; margin: 10px 0;"> <pre>                     usecaseDiagram                         ObrirTreball --&gt; VisualitzarPractica : &lt;&lt;include&gt;&gt;                     </pre> </div> </li> <li>5. Es comença a continuar amb el treball, de la mateixa manera que el cas d'ús [Crear treball]. Per tant             <div style="text-align: center; margin: 10px 0;"> <pre>                     usecaseDiagram                         ObrirTreball --&gt; CrearTreball : &lt;&lt;include&gt;&gt;                     </pre> </div> </li> </ol>
<p>Actors:</p>	<p>Alumne</p>
<p>Precondició:</p>	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a alumne.</li> <li>2. Han d'existir treballs</li> </ol>
<p>Postcondició:</p>	<p>Es va desant a la base de dades el treball que s'està modificant, el qual està associat a una pràctica que un alumne està fent.</p>

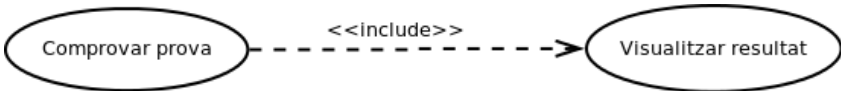
Gestionar execució

CAS D'ÚS: EXECUTAR TREBALL

<p>Funcionalitat:</p>	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Executar treball”</li> <li>2. El sistema captarà el codi escrit per l'usuari i el compilarà i executarà.</li> <li>3. Es visualitzarà el resultat de la compilació</li> </ol>
-----------------------	--

Actors:	Alumne
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a alumne.</li> <li>2. Visualitzar una pràctica.</li> <li>3. Visualitzar el treball associat a eixa pràctica.</li> </ol>
Postcondició:	No genera cap canvi al sistema.

CAS D'ÚS: COMPROVAR PROVA

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Comprovar prova”</li> <li>2. Es visualitzen les proves associades a eixa pràctica.</li> <li>3. Es selecciona la prova a comprovar.</li> <li>4. El sistema captarà el codi escrit per l'usuari.</li> <li>5. Crearà un arxiu amb el codi escrit per l'usuari i la prova associada.</li> <li>6. Aquest arxius es compila i executa.</li> <li>7. Es visualitza el resultat</li> <li>8. Es visualitza el resultat que ha de donar. Per tant aquest cas d'ús utilitza altre cas d'ús:</li> </ol>  <pre> graph LR     A([Comprovar prova]) -.-&gt; &lt;&lt;include&gt;&gt;  B([Visualitzar resultat])     </pre> <ol style="list-style-type: none"> <li>9. L'alumne els compara.</li> </ol>
Actors:	Alumne
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a alumne.</li> <li>2. Visualitzar una pràctica.</li> <li>3. Visualitzar el treball associat a eixa pràctica.</li> </ol>

Postcondició:	No genera cap canvi al sistema.
---------------	---------------------------------

## CAS D'ÚS: VISUALITZAR RESULTAT

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Visualitzar resultat”</li> <li>2. Es visualitza les proves associades a la pràctica.</li> <li>3. Es seleccionar una prova.</li> <li>4. Es visualitza el resulta que cal obtenir si li passem eixa prova al nostre codi</li> </ol>
Actors:	Alumne
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a alumne.</li> <li>2. Visualitzar una pràctica.</li> </ol>
Postcondició:	No genera cap canvi al sistema.

## CAS D'ÚS: VISUALITZAR SOLUCIÓ

Funcionalitat:	<ol style="list-style-type: none"> <li>1. S'activa en prémer el botó “Visualitzar solució”</li> <li>2. Es visualitzen les pràctiques.</li> <li>3. Es selecciona una pràctica.</li> <li>4. Es visualitza la solució.</li> </ol>
Actors:	Alumne
Precondició:	<ol style="list-style-type: none"> <li>1. Haver iniciat sessió com a alumne.</li> <li>2. Visualitzar una pràctica.</li> <li>3. La solució ha d'estar visible per als alumnes.</li> </ol>
Postcondició:	No genera cap canvi al sistema.



### 3.5.4.- Requisits no funcionals

Seràn aquells atributs que especifiquen uns criteris que es poden utilitzar per a comprovar o jutjar les operacions de l'IDE al núvol. Per tant no descriuran informació a desar ni funcions a realitzar. Els requisits no funcionals que presenta aquesta aplicació són:

- S'han d'executar els programes en el servidor de forma segura per a que no perille la integritat de la informació emmagatzema en aquest
- Les interfícies han de ser senzilles i intuïtives.
- L'accés a l'entorn web s'ha de poder fer des de qualsevol dispositiu, independentment de la plataforma i del navegador utilitzat.
- Ha de poder suportat moltes connexions alhora sense que afecte al seu rendiment.

## 3.6.- DISSENY DE L'ENTORN WEB DESTINAT A LA PROGRAMACIÓ EN C++

### 3.6.1.- Introducció

Una volta hem obtingut a les fases anteriors una descripció del model del sistema en forma de casos d'ús, passarem a fer una representació estàtica de l'anàlisi d'aquest. Utilitzarem el llenguatge de modelatge UML per a representar el sistema des del punt de vista de la anàlisi, sense tenir en compte la tecnologia que utilitzarem en la seua implementació. En aquesta representació farem servir un diagrama de classes UML on cada classe representarà un concepte del món real de l'IDE al núvol que volem desenvolupar, així com les seues relacions. Seguidament, farem una descripció de les interfícies que utilitzaran els usuaris de l'aplicació web. Aquestes serviran per a que els clients de l'aplicació puguin comunicar-se amb aquesta i així poder executar les accions relacionades amb el seu perfil. Per últim farem una descripció de la forma en que es provarà, compilarà i executarà una pràctica escrita en C++. Aquesta tasca correspon a la més important que s'ha de desenvolupar a l'aplicació web, i per tant, caldrà indicar les passes que cal donar des de que un alumne escriu un pràctica en C++ fins que la prova, compila i executa.

### 3.6.2.- Diagrama estàtic d'anàlisi

El punt de vista de la informació que volem representar a l'IDE al núvol vindrà donat per les següents classes i relacions:

#### **Usuari**

Els usuaris seran els actors que interaccionen amb el sistema. Estaran enregistrats al sistema i hauran de tenir un nom d'usuari i una contrasenya per a poder entrar-hi. Aquesta classe serà una generalització dels perfils d'usuari que poden accedir a l'aplicació web. Aquest són:

- Professor

Aquest usuari serà el responsable d'un o més grups d'alumnes. Proposarà les pràctiques que ha de fer cada grup, les proves a realitzar i els resultats que cal obtenir per a cadascuna d'elles. Per últim donarà una solució oficial a cada pràctica.

- Alumne

Usuari que accedirà al sistema per a veure les pràctiques que ha de fer. Provarà, compilarà i executarà el codi C++ que escriu a l'IDE al núvol. També tindrà un espai de treball on estaran els treballs que està desenvolupant.

- Administrador

Usuari que s'encarrega de gestionar els usuaris que poden utilitzar l'aplicació i els grups que hi hauran a cada curs escolar.

#### **Grup**

Identifica els grups que hi ha al curs escolar actual. Un grup tindrà un professor com a responsable i estarà format per molts alumnes. El professor assignarà les pràctiques que han de fer els alumnes del seu grup. Cada grup vindrà identificat per un codi i una descripció.

## Practica

Enunciat del problema que cada alumne ha de resoldre en C++. S'identifica pel seu codi i títol. Una pràctica potser assignada a un o més grups. Cadascuna d'elles tindrà associada una o més proves que l'alumne ha de realitzar per a comprovar el resultat adequats. El professor que ha creat la pràctica li assignarà una solució oficial. Tota pràctica ha de tenir una solució associada per a poder presentar a l'alumne el resultat que ha d'obtenir.

## Solució

Codi C++ que proposa el professor per a una pràctica. Associada a una pràctica.

## Prova

Funció main() que el professor proposa a cada pràctica per a que l'alumne comprove el correcte funcionament del seu treball. Una prova està associada com a molt a una practica mentres que a una pràctica se li poden assignar moltes proves.

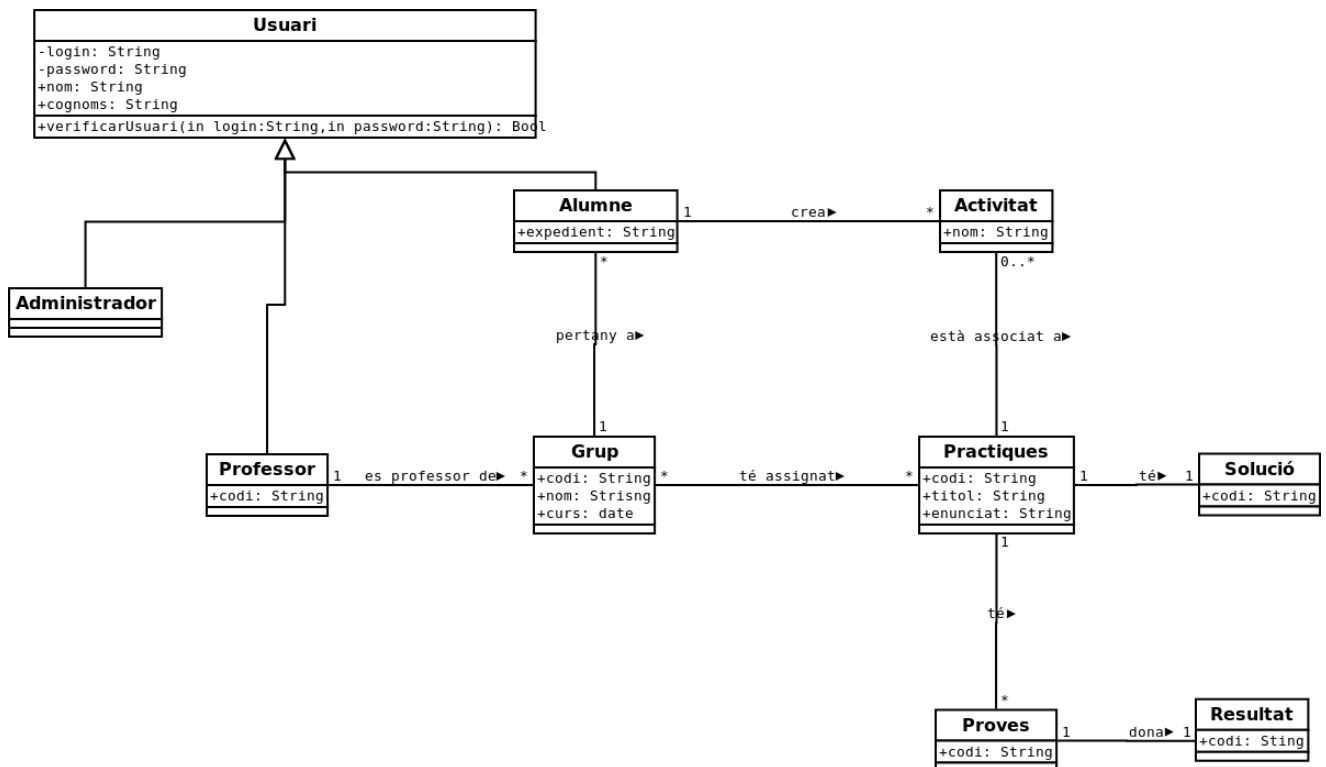
## Resultat

L'execució de la solució d'una pràctica per a cada prova dona un resultat. S'identifica per la pràctica i prova realitza.

## Treball

Cadascun del arxius fets per un alumne per a resoldre cada pràctica. Cada treball està associat a l'enunciat d'una pràctica, mentres que un alumne pot haver realitzat molts treballs o cap. En un principi, la idea és que cada treball estiguera associat a una pràctica, i una pràctica a un o cap treball (si encara l'alumne no ha començat a desenvolupar-lo). Si tenim en compte que una pràctica pot contenir la realització de diferents activitats, aleshores farem el següent plantejament: una pràctica està associada a cap o molts treballs (una pràctica de vectors pot contenir diferents activitats per practicar amb l'objecte vector) i un treball (que en aquest cas correspon a cadascuna de les activitats) correspon a una pràctica. Per tant, aquesta classe queda millor definida amb el nom d'**Activitat**.

El diagrama de classes quedarà de la següent forma:



### 3.6.3.- Interfícies

Les interfícies representen el mitjà pel qual els usuaris de l'aplicació web es comunicaran amb aquesta. Com que la construcció d'aquest IDE es basarà amb l'arquitectura client-servidor, les interfícies es visualitzaran mitjançant un navegador. A l'estudi de les tecnologies IDE vam obtenir els diferents espais que tindrà l'aplicació i algunes de les accions o característiques que proporcionarà a l'usuari per a poder interactuar amb l'eina web. La divisió en dos espais (alumne i professor) ens indica que també caldrà dissenyar-les atenen als espais que s'estan utilitzant. A més, amb la introducció de l'actor administrador a l'apartat on es descriuen els casos d'ús, s'afegirà un nou espai dedicat a aquest per a que pugui administrar l'aplicació web. En tot cas, com que moltes de les funcionalitats que proporcionaràn algunes interfícies seran comunes, podran ser utilitzades tan en un espai com en un altre diferent. Aquestes les podem incloure dins d'un espai global.

Altre aspecte a tenir en compte en la descripció i construcció de les interfícies serà que aquestes siguin simples, intuïtives i fàcils d'utilitzar, característiques que ens faciliten la consecució d'alguns dels objectius proposats al pla de treball que vam fer inicialment.

Per últim, també haurem de considerar que aquestes s'han de poder visualitzar a qualsevol tipus de navegador, independentment de la seua versió (objectiu propi de la planificació de treball). La utilització de components JQuery ens podrà ajudar a obtenir interfícies compatibles amb la majoria dels navegadors. En tot cas, caldrà fer un disseny el més estàndard possible que ens pugui

evitar aquests problemes.

A continuació, passarem a fer una descripció gràfica de les diferents interfícies que contindrà l'IDE al núvol per posar en pràctica l'estudi del llenguatge C++ en estudiants de batxillerat i cicles formatius als diferents espais identificats anteriorment:

## Espai global

- La primera interfície que tot usuari del sistema haurà de visualitzar haurà de sol·licitar-li el seu nom i contrasenya. Si l'usuari no està enregistrat ho indicarà.

The image displays two side-by-side screenshots of the 'Cloud Code Test IDE' login interface. Both screenshots have a light yellow background and a title bar at the top that reads 'Cloud Code Test IDE'. The left screenshot shows a standard login form with two input fields: 'Nom d'usuari' and 'Contrasenya', and a button labeled 'Entrar' below them. The right screenshot shows the same form, but with a red message 'Usuari no registrat' displayed above the input fields, indicating that the user is not registered.

- La resta de tasques que apareixen al sistema genèric del diagrama de casos d'usos es troben incloses a les interfícies dels espais alumne, professor i administrador.

## Espai alumne

- Aquesta interfície visualitzarà les pràctiques que el professor ha proposat al grup, les activitats que l'alumne tinga fetes de cada pràctica i les proves i resultats que haurà d'obtenir per a cadascuna d'elles. Amb aquesta interfície l'alumne podrà executar els casos d'ús relacionats amb la gestió de l'edició i execució de cadascuna de les activitats proposades a les pràctiques.

# Cloud Code Test IDE - ALUMNE

---

Nom de l'alumne
Nom del grup
Nom del professor

---

PRÀCTICA-1
PRÀCTICA-2
PRÀCTICA-3

## Enunciat de la pràctica seleccionada

### TREBALLS

- Activitat-1
- Activitat-2
- Activitat-3

Crear
Desar
Eliminar
Solució

Eixir

### PROVES

- Cap
- Prova-1
- Prova-2
- Prova-3

### Codi

Codi C++ de l'activitat 2 de la pràctica 1

Executar

### Execució

Compilació i execució de l'activitat 2 de la pràctica 1 amb el main() de la prova 2. Si a proves elegim cap, s'executarà amb el main() que escriga l'alumne al codi

### RESULTAT

Resultat que l'alumne ha d'obtenir amb l'execució de la Prova-2  
Si a proves elegim cap, no es visualitzarà res

## Espai professor

- Els casos d'usos relacionats amb l'actor professor es classifiquen en aquells relacionats amb la gestió de les pràctiques, gestió de la solució i en la gestió de la prova. El professor farà cadascuna d'aquestes activitats a una única interfície, la qual visualitza de dalt a avall cadascuna de les gestions que ha de fer. A més, podem veure que al final de la interfície el professor pot emmagatzemar el resultat obtingut de l'execució de la seua solució a cadascuna de les proves que ha afegit, i que l'alumne haurà de comprovar. Aquesta gestió dels resultats que ha de fer el professor no s'han descrit al casos d'ús fets a la fase anterior. Caldrà fer una rectificació en eixe apartat.

**Cloud Code Test IDE - PROFESSOR**

Nom del professor

PRÀCTICA-1 PRÀCTICA-2 PRÀCTICA-3

Crear  
Modificar  
Eliminar

El professor escriu l'enunciat de la pràctica

GRUPS  
Grup-1  
Grup-2  
Grup-3

Afegir

PRÀCTICA - GRUP  
Pràctica-2 : Grup-1  
Pràctica-2 : Grup-2

Extraure

SOLUCIÓ

Afegir  
Eliminar  
Modificar  
Publicar

Solució en C++ de la pràctica

EXECUCIÓ

Executar

Resultat solució + prova

PROVES  
Prova-1  
Prova-2  
Prova-3

Funció main de la prova 2

Afegir  
Eliminar  
Modificar

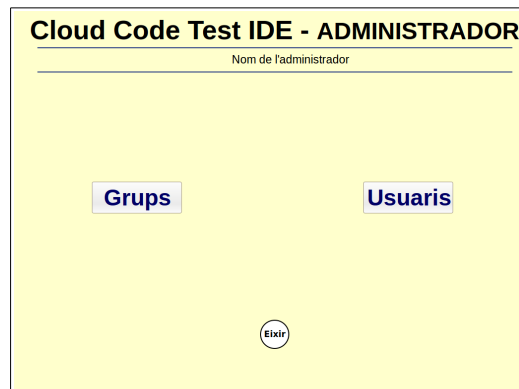
Desar resultat

Eixir

## Espai administrador

- Les tasques que realitza aquest actor estan relacionades amb la gestió del usuaris i grups. Per tant, l'administrador s'encarregarà de crear, modificar i/o eliminar aquests tipus d'usuaris. Altres de les tasques que haurà de fer i que no s'han reflectit al diagrama de casos d'ús són les associades amb l'assignació a cada grup dels professors i alumnes. Les interfícies creades integren totes aquestes accions. En tot cas, l'administrador accedirà a un primera pantalla per accedir-hi a la resta de

gestions que podrà fer.

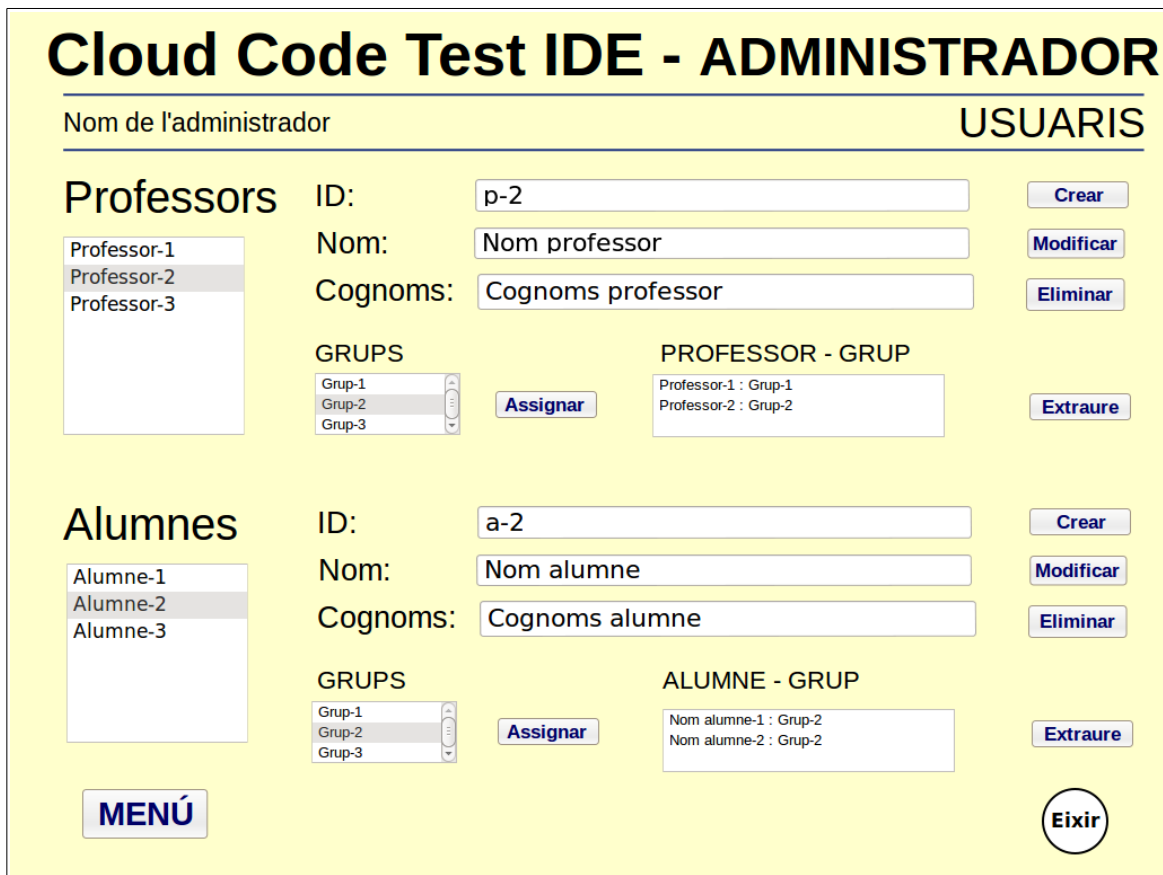


La següent interfície proporciona a l'administrador les tasques relacionades amb la gestió dels grups.

The screenshot shows a yellow background with the title "Cloud Code Test IDE - ADMINISTRADOR" at the top. Below the title is a horizontal line and the text "Nom de l'administrador" on the left and "GRUPS" on the right. Below this, there are three buttons: "Crear", "Modificar", and "Eliminar". On the left, there is a "Llistat" section with a list of groups: "Grup-1", "Grup-2", and "Grup-3". On the right, there are three input fields: "Codi:" with the value "GR2", "Nom:" with the value "Grup-2", and "Descripció:" with the value "Alumnes 1r BAT". At the bottom, there are two buttons: "MENÚ" and "Eixir".



La següent interfície proporciona a l'administrador les tasques relacionades amb la gestió dels usuaris.



### 3.6.4.- Execució i compilació

L'IDE al núvol que volem desenvolupar es basarà en un sistema client-servidor. Quan es connecta un alumne podrà compilar i executar el codi C++ que escriga, a més de poder comprovar la seua execució a partir d'unes proves proposades pel professor. Per tant, cal determinar on es realitzarà la compilació i execució, és a dir, al costat del client o al costat del servidor. En tot cas, l'anàlisi fet als anteriors punts de la memòria ens dona una informació molt vàlida per a decidir on realitzar aquesta tasca de prova, compilació i execució d'una pràctica en C++.

Uns dels objectius proposats al principi del projecte ens diu que aquesta eina haurà de proporcionar a l'alumnat l'aprenentatge del llenguatge C++ a qualsevol plataforma informàtica i ha de ser independent del recursos informàtics que tinga disponibles. Una compilació al costat del client haurà d'assegurar que els recursos consumits siguin suficients per a que aquesta siga efectiva. La varietat de característiques de maquinari des d'on els alumnes poden accedir-hi i la cada volta més estesa utilització de clients lleugers al centres de secundària, indiquen que mai podrem conèixer si el dispositius utilitzats per accedir-hi proporcionen els recursos necessaris per a que l'alumne pugui treballar amb el llenguatge C++. A més, com que es basa en un sistema

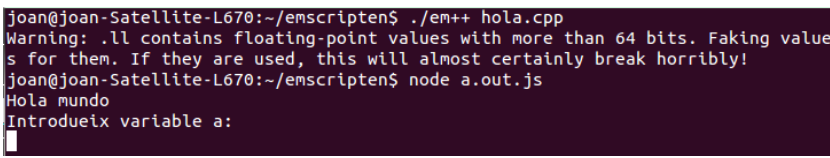
client-servidor, el mitjà pel qual s'haurà de provar, compilar i executar el codi C++ serà el navegador. Per tant, aquestes tasques s'hauran de traduir a un llenguatge que puga ser interpretat per aquest, com per exemple Javascript. Això també implica que el navegador del client haurà de tenir instal·lat el plug-in corresponent per a poder interpretar l'arxiu obtingut en la prova, compilació i execució d'una pràctica en C++. En canvi, executant l'aplicació web al costat del client implica una major seguretat en els arxius que hi ha al servidor, no cal implementar una seguretat en la transmissió de les dades per la xarxa i no hi haurà cap espera en obtenir els resultats ja que aquests el proporciona el mateix client. En aquest cas, farem predominar més els objectius inicials del projecte que les avantatges de l'execució al costat del client.

Realitzar la prova, compilació i execució de codi C++ al costat del servidor ens dona una sèrie d'avantatges que faciliten la consecució dels objectius del projecte. En un principi coneixerem les característiques de maquinari del servidor i es tindran en compte en el desenvolupament del programari, sobre tot en la seua complexitat. Com que la prova, compilació i execució es realitza al costat del servidor pot donar-se el cas de que aquest es sobresature. Així que haurem de veure un servidor capaç d'atendre moltes peticions i que aquestes es mantinguen obertes per a que els alumnes puguen treballar amb el llenguatge C++. Node.js ens pot donar una solució a aquest problema, ja que permet mantenir moltes connexions obertes i esperant. Aquest entorn Javascript del costat del servidor utilitza un model asíncron i dirigit per esdeveniments, cosa que fa possible mantenir en un mateix fil moltes connexions<sup>4</sup>. També aprofitarem la seua capacitat d'executar aplicacions escrites en Javascript per a executar el resultat obtingut en la compilació de les pràctiques dels alumnes. Anteriorment vam comentar que una manera de poder executar qualsevol aplicació a qualsevol plataforma és amb la utilització d'un codi intermedi que ens facilite la seua posterior traducció a un llenguatge més relacionat amb la web, com per exemple Javascript. Compiladors com Clang ens ajudaran a traduir una aplicació escrita en C++ al codi intermedi LLVM, mentres que emscripten realitzarà la traducció d'LLVM a Javascript. A continuació Node.js l'executarà i enviarà el resultat obtingut al navegador de l'alumne que li havia fet eixa petició.

Javascript és un llenguatge orientat a objectes i orientat a esdeveniments que es centra en la descripció dels objectes i en l'escriptura de funcions que responguen a moviments de ratolí, prémer tecles, obrir i tancar finestres, carregar una pàgina, etc .. i moltes altres ocurrències. Aquesta forma de treballar pot ser un problema quan utilitzem Emscritpen per a compilar i traduir a Javascript aquelles funcions o mètodes de C++ que requereixen interacció amb l'usuari mitjançant el teclat, com per exemple la utilització de *cin* per a llegir dades des de la consola. En canvi a Javascript, la interacció amb l'usuari requereix d'una finestra on l'usuari podrà introduir la dada sol·licitada, per exemple amb la utilització de la funció *prompt*. Per comprovar el que passa, hem compilat el següent exemple des del terminal:

---

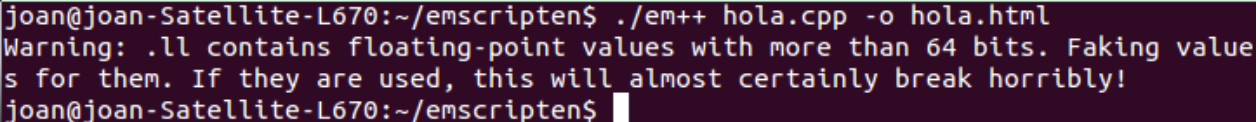
<sup>4</sup> En una activitat normal podria mantenir fins a 25.000 clients al mateix temps sense apenes retard a les respostes.

<b>hola.cpp</b>	<b>Eixida obtinguda</b>
<pre>#include&lt;iostream&gt; using namespace std; int main() {     int a;     cout &lt;&lt; "Hola mundo" &lt;&lt; endl;     cout &lt;&lt; "Introdueix variable a:" &lt;&lt; endl;     cin &gt;&gt;a;     cout &lt;&lt; "La variable a val: " &lt;&lt; a;     return 0; }</pre>	 <pre>joan@joan-Satellite-L670:~/emscripten\$ ./em++ hola.cpp Warning: .ll contains floating-point values with more than 64 bits. Faking values for them. If they are used, this will almost certainly break horribly! joan@joan-Satellite-L670:~/emscripten\$ node a.out.js Hola mundo Introdueix variable a: █</pre> <p>L'arxiu Javascript a.out.js es genera però amb un warning. L'execució d'aquest script a node.js es queda penjada.</p>

En aquest exemple l'execució del programa en C++ hauria de demanar a l'usuari un valor per a després visualitzar-lo. Si visualitzem l'script Javascript generat podem veure les línies on es realitza la introducció de les dades mitjançant la funció *prompt*. Al codi a.out.js apareixen les línies on es realitza la lectura per teclat utilitzant la funció *prompt*:

```
if (typeof window !== 'undefined' &&
    typeof window.prompt == 'function') {
    // Browser.
    result = window.prompt("Input: ");
    if (result === null) result = String.fromCharCode(0);
```

Aquesta funció necessita executar-se en un context HTML per a visualitzar-la en un navegador. Des d'aquí l'usuari podrà introduir el valor de la variable que se li demana. Per tant, altra possibilitat consisteix en poder compilar el codi C++ a un document HTML que portara incrustat l'script en Javascript que realitza l'execució d'eixe codi. Així, el compilador Emscripten generarà un arxiu HTML amb codi Javascript incrustat. Aquesta funcionalitat la pot fer amb el següent comandament:



```
joan@joan-Satellite-L670:~/emscripten$ ./em++ hola.cpp -o hola.html
Warning: .ll contains floating-point values with more than 64 bits. Faking values for them. If they are used, this will almost certainly break horribly!
joan@joan-Satellite-L670:~/emscripten$ █
```

En aquest cas, l'entorn node.js no executaria un script en Javascript si no que treballaria com un servidor de pàgines web. Aquestes serien rebudes pels navegadors que han fet les peticions, els qual executarien l'script incrustat. Aquesta acció es portaria a terme sempre i quan el navegador

tinga habilitada aquesta funcionalitat i el punt d'accés de l'alumne dispose dels recursos necessaris. Aquesta forma de treballar al costat del client no la podem donar com a vàlida ja que no ajuda a la consecució dels objectius del nostre IDE al núvol.

Altra possibilitat que podríem contemplar és traduir la totalitat del codi HTML generat per emscripten a un script Javascript. D'aquesta manera l'entorn node.js l'executaria i estaríem treballant del costat del servidor. Per contra, continuariem tenint el problema en la lectura de dades per teclat. Per tant, aquesta serà una limitació que tindrà el nostre sistema

En definitiva, aquesta IDE al núvol es basarà en un sistema client-servidor on la compilació i l'execució del codi C++ es farà al costat del servidor. El funcionament genèric d'aquesta aplicació web vindrà descrita pels següents passos:

- Un client fa una petició de prova, compilació i execució del codi que ha escrit a l'aplicació web.
- El servidor rep aquesta petició i realitza els següents passos:

1. Agafa el codi rebut via http i li afegeix la prova que ha seleccionat el client.

2. El codi resultant l'emmagatzema al seu disc amb l'extensió .cpp:

```
codi.cpp
```

3. S'executa emscripten al servidor per a compilar i obtenir l'script en Javascript de l'arxiu emmagatzemat al disc.

```
em++ codi.cpp
```

4. Si la compilació ha donat un error aquest es captura i s'emmagatzema com a resposta.

5. Si la compilació s'ha realitza amb éxit, l'script de Javascript resultant s'executa a l'entorn node

```
node a.out.js
```

i es captura l'execució de l'script a node.js i s'emmagatzema com a resposta.

- El servidor respon al client enviant-li la resposta que ha emmagatzemat

## 4.- IMPLEMENTACIÓ

### 4.1.- DESENVOLUPAMENT INICIAL DE L'APLICACIÓ WEB

#### 4.1.1.- Introducció

En aquest apartat anem a començar a desenvolupar el producte web a partir de les eines tecnològiques elegides a les fases anteriors, i sempre tenint en compte els requisits, dissenys i interfícies aconseguides. Abans de començar a desenvolupar totes les funcionalitats que ha de proporcionar l'aplicació, caldrà primer de tot tenir clar la manera en que es compilarà i executarà el codi C++. Aquesta tasca anem a considerar-la com a primordial i bàsica abans de començar a implementar qualsevol altra funcionalitat no relacionada amb l'objectiu principal que tot IDE de C++ al núvol ha de fer: compilar i executar codi C++.

Abans de començar a indicar les tasques que anem a desenvolupar en aquest inici d'implementació, hem d'indicar que a les fases anteriors ja vam instal·lar el compilador emscripten i tot el programari vinculat a ell, així com node.js. Per tant, a continuació començarem a utilitzar node.js i emscripten per a construir l'aplicació web.

#### 4.1.2.- Creació del lloc web

Primerament construirem l'estructura de l'espai web que albergarà l'IDE de C++. Per a facilitar-nos la feina utilitzarem Express<sup>5</sup>, un framework web per a node.js. La seua instal·lació es realitza via npm des de node :

```
joan@joan-Satellite-L670:~\$ sudo npm install -g6 express
```

Seguidament crearem l'estructura bàsica (l'esquelet) d'un lloc creat amb express, indicant el nom que tindrà en nostre projecte:

---

5 Un framework ens assegura un codi estable i segur ja que es troben provats i testejats. A més, també ens ajuda a desenvolupar projectes web's d'una forma més ràpida. Amb Express no tindrem que fer l'aplicació des de nou i comptarem amb patrons comuns a tota aplicació web com poden ser els routings, capes de vista, etc..

6 Amb l'opció -g farem la instal·lació de forma global. D'aquesta manera el tindrem disponible des de qualsevol lloc del sistema. En altre cas, tindrem que indicar la ruta on es troben els executables d'express.

```
joan@joan-Satellite-L670:~$ express ide
create : ide
create : ide/package.json
create : ide/app.js
create : ide/public
create : ide/public/javascripts
create : ide/public/images
create : ide/public/stylesheets
create : ide/public/stylesheets/style.css
create : ide/routes
create : ide/routes/index.js
create : ide/routes/user.js
create : ide/views
create : ide/views/layout.jade
create : ide/views/index.jade

install dependencies:
$ cd ide && npm install

run the app:
$ node app
```

Amb el projecte ide creat, se'ns indica que cal instal·lar les dependències per al projecte:

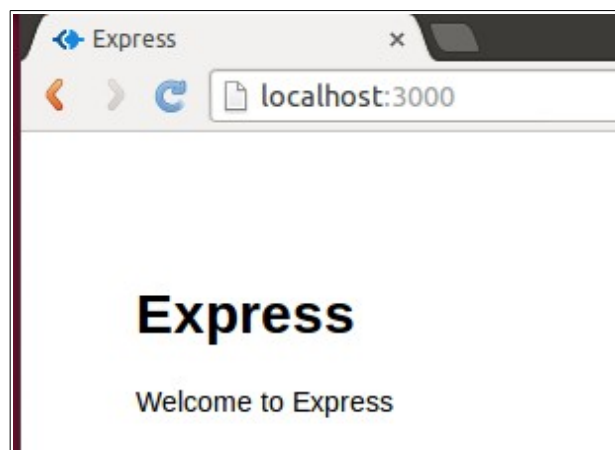
```
joan@joan-Satellite-L670:~$ cd ide && npm install
```

També ens diu, la manera en que el podem executar l'aplicació web creada, una volta ens

```
joan@joan-Satellite-L670:~/ide$ node app.js
Express server listening on port 3000
```

situem al directori on es troba el lloc web:

Ara ja tenim el servei web disponible per a la nostra aplicació web, gràcies al framework Express. A continuació executem el navegador i introduïm l'adreça amb el port on està escoltant:



#### 4.1.3.- Estructura del lloc web

El lloc web creat per Express ens proporciona una estructura de carpetes on poder generar tota la lògica de negoci i vistes que haurà de tenir l'IDE al núvol. A continuació apareixen les carpetes i arxius que ha generat per defecte Express:



- **app.js** --> Arxiu utilitzat per a iniciar l'aplicació. Conté la informació de configuració per a l'aplicació.
- **package.json** --> Dona informació a l'aplicació, incloent-hi les dependències que han de ser instal·lades per a que funcione.
- **node\_modules** --> Conté tots els mòduls definits en package.json i que hagen sigut instal·lats.
- **public** --> Directori públic que serveix l'aplicació web. Al seu interior s'inclouen els css (fulls d'estil), js (javascript) i imatges. Mai ha d'incloure lògica. D'aquesta manera es segueix el MVC com a patró.
- **routes** --> Directori que conté les declaracions de les rutes per a enrutar les nostres pàgines.
- **views** --> Directori on es defineix el disseny i aspecte de l'aplicació. Dins del directori views hi ha dos arxius amb extensió jade<sup>7</sup> que fan referència al motor de plantilla que utilitza Express.

#### 4.1.4.- Aplicació bàsica

Com hem dit a la introducció, el primer que haurem de fer és implementar la tasca principal que ha d'executar l'aplicació web. Aquesta consisteix bàsicament en agafar el codi C++ que un alumne escriu a un apartat del navegador i obtenir dins de la mateixa pàgina el resultat obtingut. Aquesta implementació la farem seguint el funcionament genèric que hem descrit a l'apartat on hem

---

<sup>7</sup> Jade és un motor de plantilles que utilitza el sagnat per a declarar la jerarquia HTML. Express l'utilitza per a generar el codi HTML de les seues vistes.

fet l'estudi de l'execució i compilació.

Un dels arxius que haurem de modificar per a implementar l'aplicació web és l'app.js. En aquest resideixen els mòduls o arxius externs necessaris per a executar l'aplicació, on es crea el servidor i se li assigna el nom d'app, on es fa la configuració per a que l'aplicació funcione correctament (middleware<sup>8</sup>) i on s'especifiquen les rutes i el mètode HTTP per a cadascuna d'elles. Aquestes últimes són molt importants ja que si no estan ben definides no podrem realitzar les interfícies per al client. Per tant, el primer que anem a fer és especificar les rutes i els mètodes per a poder fer funcionar de forma genèrica la compilació i execució de codi C++.

## Rutes

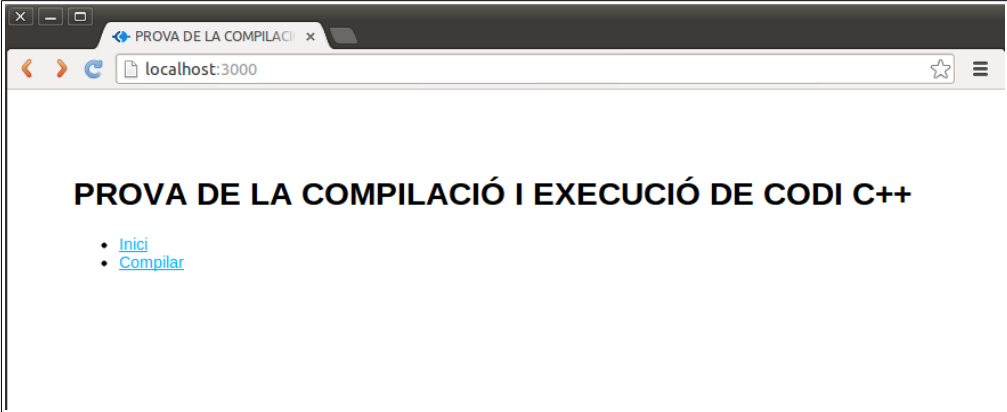
- Les rutes venen especificades de la següent forma: **app.VERB (PATH, ACCIÓ);**
- Verb representa cadascun dels mètodes de petició que es poden fer: get, post, ..
- Path l'adreça d'accés.
- Acció és allò que s'ha de fer.

Les rutes creades en aquesta aplicació inicial, amb la seua descripció i vista generada són:

---

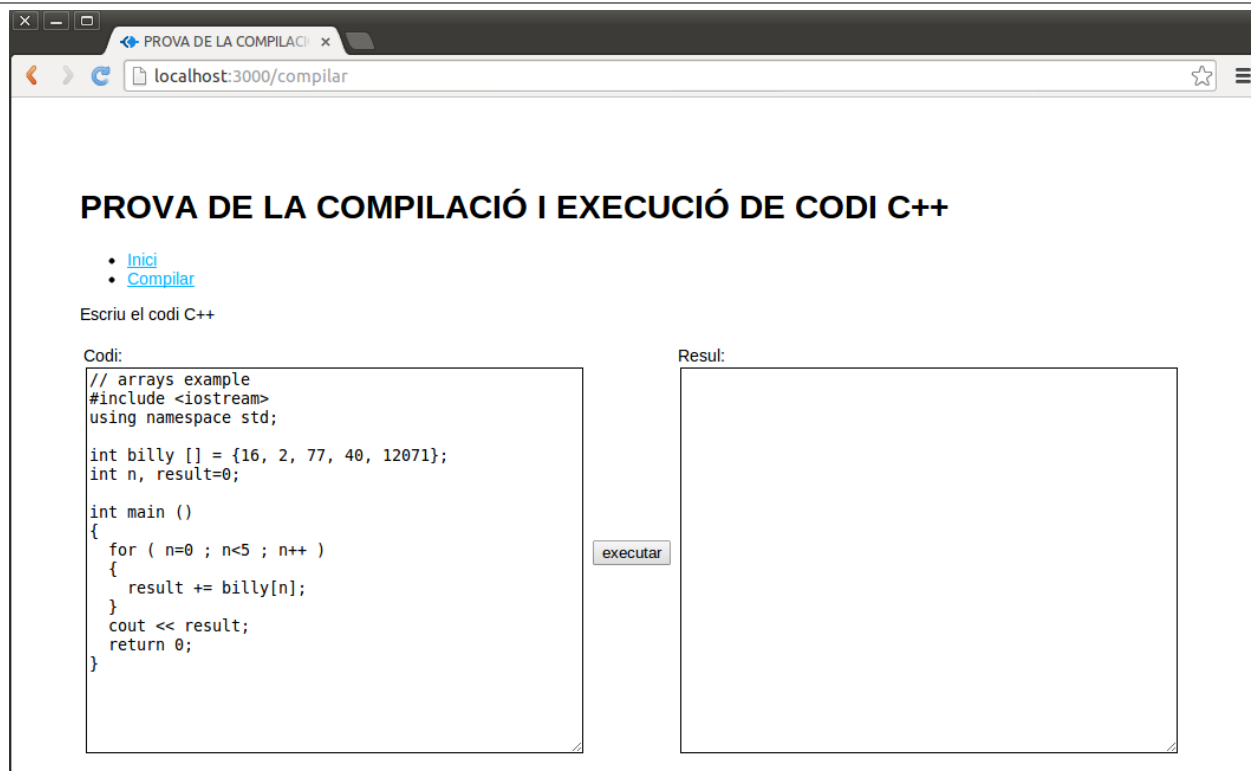
8 Programari que assisteix a una aplicació per a interactuar o comunicar-se amb altres aplicacions, programari, xarxes, maquinari i/o sistemes operatius. Aquesta capa d'abstracció facilita el treball als programadors en la generació de connexió en sistemes distribuïts. Per exemple, a la declaració es fa un ús del middleware **app.router**, el qual conté totes les rutes que hi ha definides i realitza la consulta de rutes basant-se amb la URL de la sol·licitud actual i el mètode HTTP.



RUTA: localhost:3000	DESCRIPCIÓ
<pre>app.get('/', function(req, res){     codi="";     res.render('index.jade', {codi:codi}); });</pre>	<p><i>Codi</i> representa una variable Javascript que contindrà el codi rebut del client (es troba definida després dels requeriments). Quan s'accedisca a l'arrel de l'aplicació ('localhost:3000'), es retornarà (get) el recurs sol·licitat per l'usuari. El primer que s'ha de fer és inicialitzar la variable <i>codi</i> a buit. A continuació, es crida al mètode <i>render</i> (mostrar) del paràmetre <i>res</i> passant-li el nom de la plantilla que ha de renderitzar (<i>index.jade</i>). A aquesta se li passa la variable <i>codi</i> per a que visualitze el codi C++ escrit anteriorment per l'alumne (la primera volta estarà buit).</p>
<h3>VISTA</h3> 	
<ul style="list-style-type: none"> <li>• L'enllaç <u>Inici</u> ens torna a la mateix ruta inicial: '/'</li> <li>• L'enllaç <u>Compilar</u> en porta a la vista on poder compilar el codi C++.</li> </ul>	

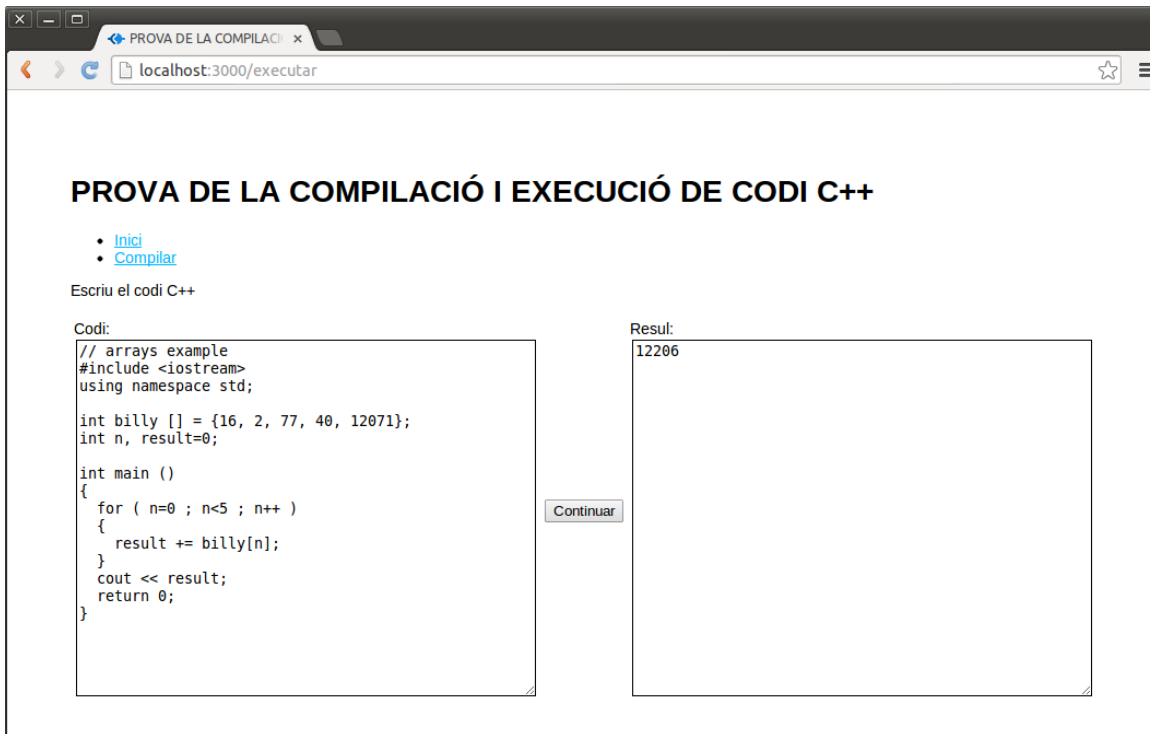
RUTA: 'localhost:3000/compilar'	DESCRIPCIÓ
<pre>app.get('/compilar',function(req, res){     res.render('compilar:jade',{codi:codi, resultat:resultat}); });</pre>	<p>El funcionament és el mateix que l'anterior. En canvi, a la plantilla 'compilar:jade' li passem les variables <i>codi</i> i <i>resultat</i>. Aquesta última contindrà el resultat obtingut en la compilació i execució del contingut de la variable <i>codi</i>. Si és la primera volta que es compila i executa codi C+ aquestes estaran buides.</p>

**VISTA**

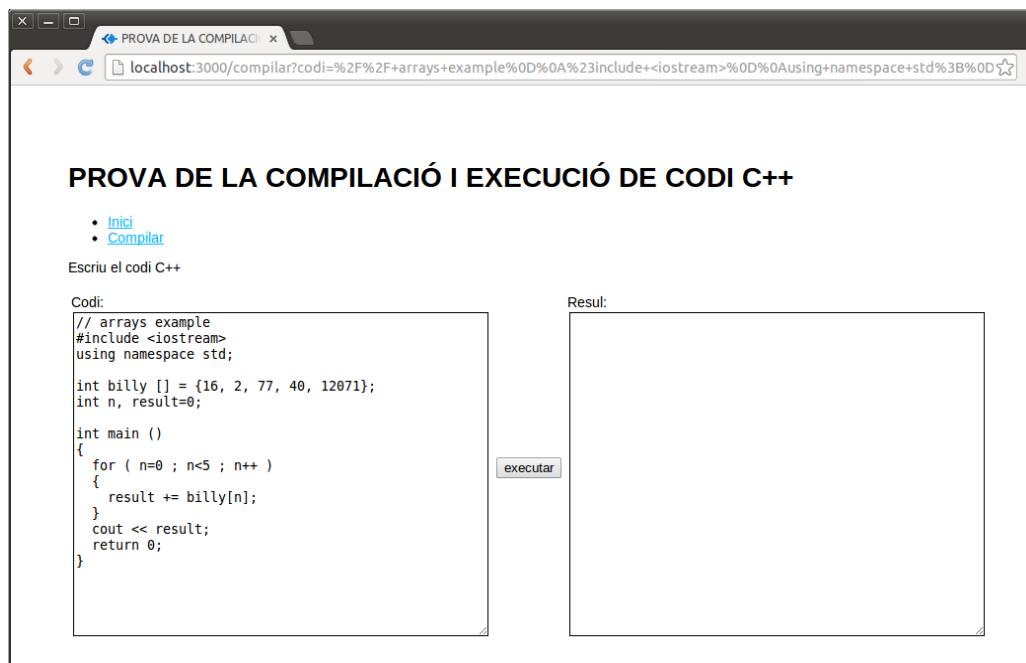


- L'alumne introdueix el codi al requadre especificat.
- A la plantilla 'compilar:jade' es crea un formulari amb dos atributs: action=/executar i method=post. El primer indica a quina ruta hem d'apuntar (/executar, que haurà d'estar creada al servidor) i amb el segon enviem (post) al servidor les dades introduïdes al formulari. En aquest es creen els textarea *codi* i *result* per a introduir text. El primer contindrà el codi escrit per l'usuari i el segon el resultat obtingut. Inicialment *result* no visualitzarà res.
- Al prémer el botó **executar** s'envia el formulari.

<p><b>RUTA: 'localhost:3000/executar'</b></p>	<p><b>DESCRIPCIÓ</b></p>
<pre> app.post('/executar',function(req, res){    fs.writeFile('./codi/codi.cpp', req.body.codi, 'utf8', function (err) {      .....    });    var exec = require('child_process').exec;    var child1,child2;    child1 = exec("../emscripten/em++ ./codi/codi.cpp", function (error, stdout, stderr) {      if (error !== null) {        res.render('executat.jade', {codi:req.body.codi,resultat:stderr});      }else{        child2 = exec("node ./a.out.js", function (error, stdout, stderr) {          if (error !== null) {            console.log('exec error: ' + error);          }else            res.render('executat.jade', {codi:req.body.codi,resultat:stdout});          });        }      });      codi=req.body.codi;    }); </pre>	<p>Amb el <i>middleware bodyParser</i> accedim als valors passats pel formulari mitjançant l'objecte <b>req.body</b>. Primer creem l'arxiu ./codi/codi.cpp amb el valor passat al codi (req.body.codi). D'aquesta manera tindrem disponible el codi C++ al disc dur.</p> <p>A continuació llançarem un procés fill per a executar el comandament d'execució i compilació amb emscripten (indicant la ruta on es troba l'executable em++). Si ens dona error de compilació, assignarem a la variable resultat el contingut del buffer , en altre cas llancem un segon procés fill per a executar l'script obtingut a node. En tots dos casos es renderitza la plantilla 'executat.jade' per a visualitzar al formulari les variables <i>codi</i> i <i>resultat</i>.</p> <p>Emmagatzem el paràmetre codi passat a la variable <i>codi</i>, i així el tindrem disponible per si l'alumne vol rectificar-lo.</p>
<p style="text-align: center;"><b>VISTA</b></p>	

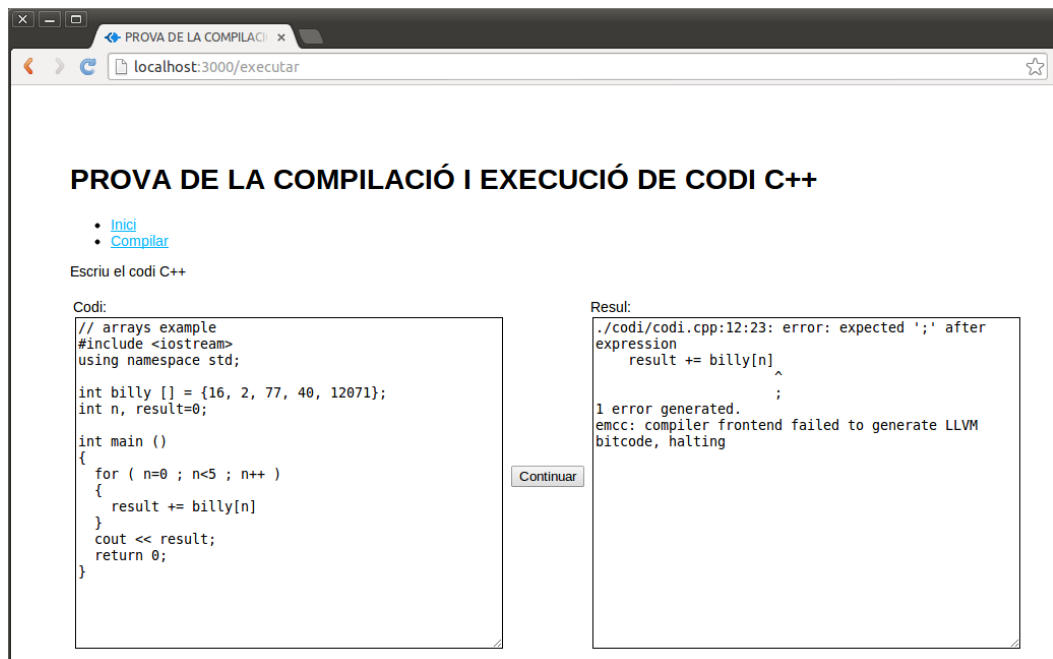


- A la plantilla 'executat.jade' hi ha el mateix formulari que 'compilar.jade'. La diferència és que els atributs del formulari tenen la següent informació: `action='/compilar'` `imethod=get`. La vista generada és:



- Com hem desat el codi de l'alumne a la variable `codi`, podem tornar a visualitzar-lo per a que l'alumne el pugui modificar si hi ha hagut errors o bé pugui continuar amb el seu desenvolupament.

- Si indiquem un codi amb error, també es visualitzarà al quadre Result l'error comés:



## Vistes

- Per a poder generar les vistes s'ha utilitzat el motor de plantilla Jade que incorpora el framework Express.
- Les vistes es troben emmagatzemades a la carpeta View.

## Espai de treball.

- En aquesta primera aproximació a l'IDE al núvol, el codi C++ enviat per l'usuari s'emmagatzema en un arxiu de nom 'codi.cpp' a la carpeta del projecte ide/codi. En canvi, l'script de Javascript a.out.js es crea a la carpeta ide. Evidentment, si realitzem la prova amb diferents usuaris a la volta segurament que alguns d'ells no obtindran el resultat esperat. Caldrà que cada usuari que es connecte a l'aplicació tinga assignat el seu espai de compilació i execució per a que no hi haja confusió amb el codi font i l'script generat.

## 4.2.- DESENVOLUPAMENT OPCIONAL DE L'APLICACIÓ WEB

### 4.2.1.- Introducció

Fins aquesta fase el producte implementat sols realitza la compilació i execució de codi C++. Aquesta és la tasca més bàsica que ha de fer l'aplicació. Per contra, encara no està implementada la lògica on cada alumne pugui visualitzar les pràctiques a fer i provar els seus codis amb les proves

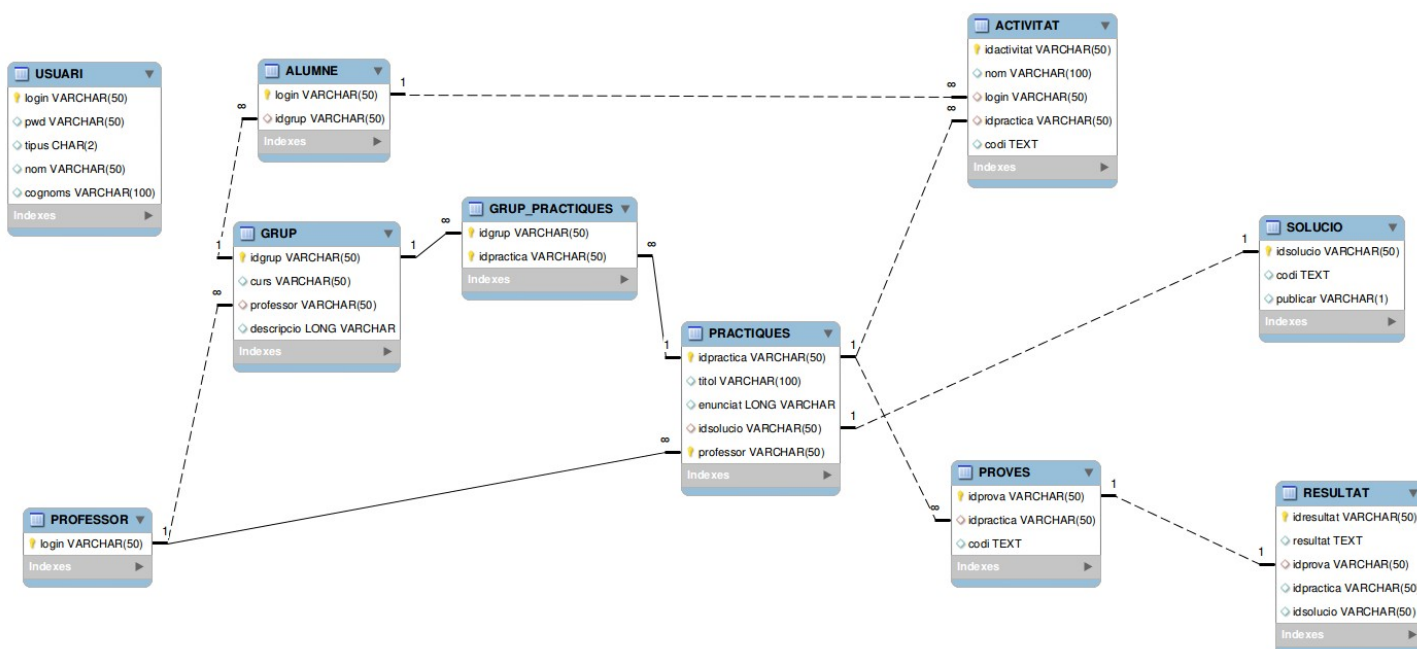
proporcionades pel professor. Tampoc un professor pot gestionar les pràctiques que cada alumne ha de fer i l'administrador encara no té els mitjans per a poder gestionar l'IDE amb els diferents professors, grups i alumnes. Al funcionament d'aquesta aplicació web també considerem aquestes tasques com a bàsiques per a que l'aplicació es pugui utilitzar dins d'un context educatiu.

Inicialment, l'objectiu d'aquesta última fase d'implementació del producte estava condicionada a les tasques realitzades a l'anterior. Com que encara no tenim implementat l'ús bàsic de l'aplicació per a que els actors (administrador, professor i alumne) puguin executar les funcionalitats que tenen associades, dedicarem aquesta etapa per a implementar-les. Els diferents passos que realitzarem seran:

- Creació de la base de dades per a fer persistent la informació.
- Creació de les vistes definides a la fase anterior.
- Implementació de la lògica.

#### 4.2.2.- Creació de la base de dades

Per a la construcció de la base de dades utilitzarem el diagrama de classes obtingut a la fase “*Disseny de l'entorn web destinat a la programació en C++*”. En primer lloc caldrà escriure l'script per a crear la base de dades amb el gestor de dades relacional mysql. Per a facilitar-nos la tasca hem utilitzat MySQL Workbench<sup>9</sup>. El disseny d'entitats i relacions creat és el següent:



<sup>9</sup> Eina visual de disseny de base de dades que integra desenvolupament de programari, administració de bases de dades, disseny de bases de dades, creació i manteniment per al sistema de bases de dades MySQL.

A partir d'aquest, s'ha creat l'script que generarà la base de dades que utilitzarà l'aplicació web per a fer persistent les dades referides als actors que interactuen en el sistema: administrador, professor i alumne. En aquest script s'ha inclòs la inserció de l'actor administrador:

- `INSERT INTO `ide`.`USUARI` (`login`, `pwd`, `tipus`, `nom`, `cognoms`) VALUES ('admin', 'admin', 'ad', 'Joan', 'Carbonell');`

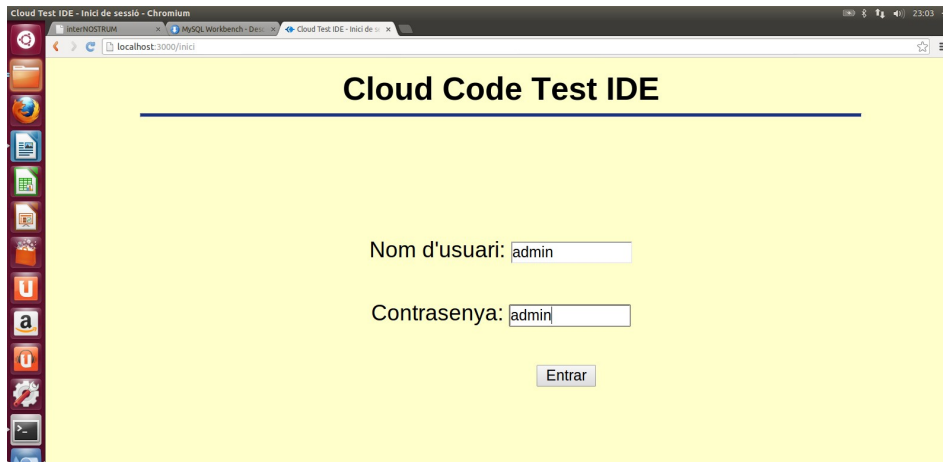
Amb aquesta eina també visualitzarem els canvis que van produint-se a la base de dades a mesura que els actors interactuen amb l'IDE al núvol. Aquesta visualització en temps real ens ajuda a poder desenvolupar tota la lògica de l'aplicació així com a resoldre els errors produïts en la utilització de les comandes d'inserció, actualització i selecció sobre les diferents entitats i relacions.

#### 4.2.3.- Creació de les vistes definides a la fase anterior.

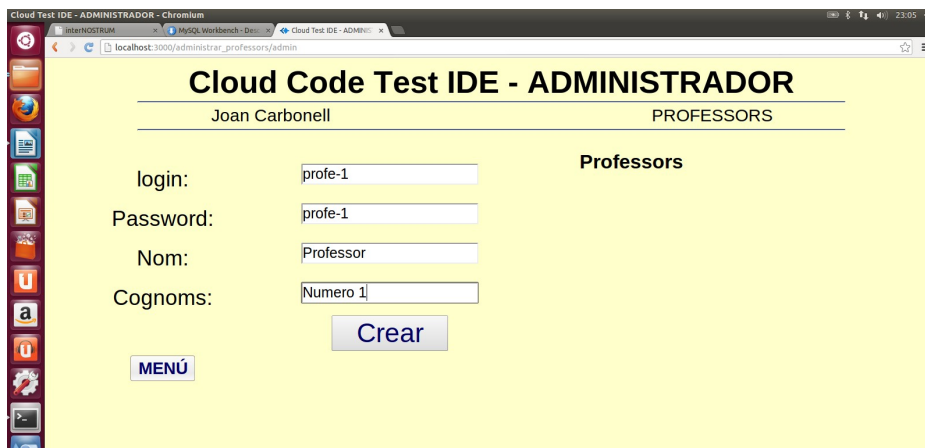
Respecte a la creació de les interfícies dissenyades a la fase “*Disseny de l'entorn web destinat a la programació en C++*”, s'han fet una sèrie de modificacions. En un principi, les tecnologies elegides per a implementar-les eren el motor de plantilles Jade i JQuery, una biblioteca Javascript. L'aprenentatge de Jade ha estat molt senzill i intuïtiu, ja que la seua sintaxis és pareguda a HTML. L'únic problema és que al basar-se amb la identació, cal tenir cura. Per contra, quan vaig començar a fer l'estudi de la tecnologia JQuery, vaig veure que caldria més temps de l'esperat. Els meus bàsics coneixements de JavaScript no m'han permès poder aprendre les funcionalitats de JQuery que m'hagueren servit per a dissenyar les interfícies dissenyades inicialment. Per tant, vaig tenir que modificar-les per a poder-les desenvolupar dins del plaç estipulat al pla de treball. Això sí, fent-les intuïtives i senzilles d'utilitzar per a que els alumnes de batxillerat i cicle formatiu puguin interactuar amb l'aplicació sense cap problema. Per veure un exemple de com han quedat, a continuació apareixen les diferents interfícies que apareixen per als actors del sistema:

## Administrador

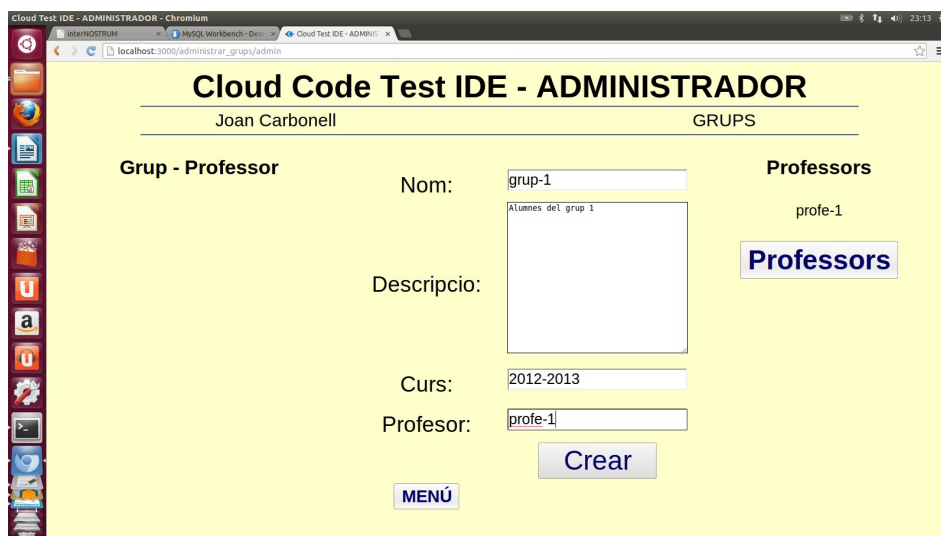
- Entrada al sistema:



- Creació d'un professor:

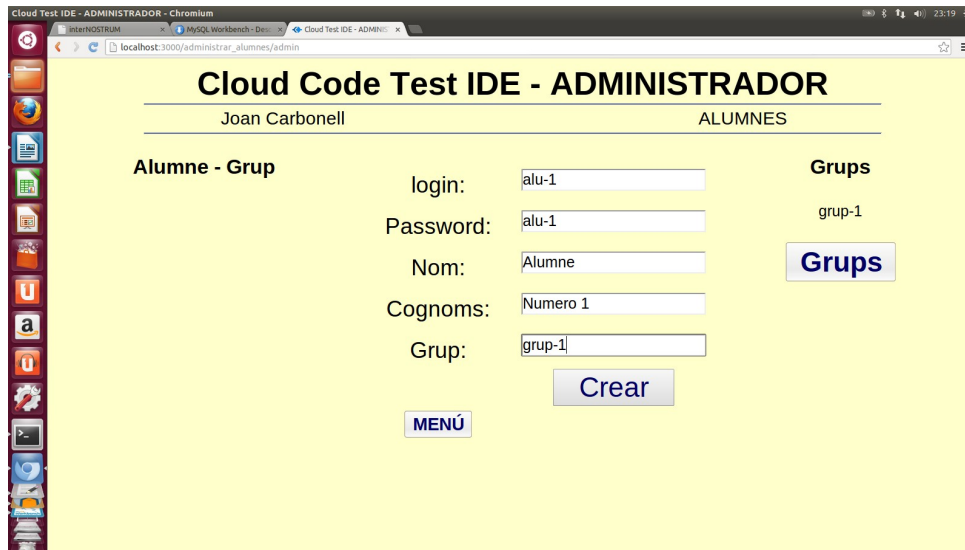


- Creació d'un grup associat a un professor:



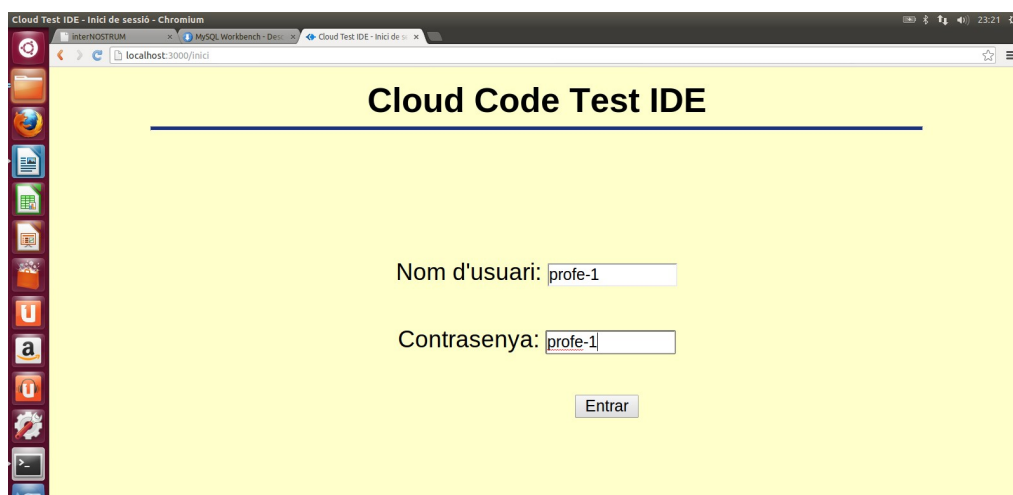


- Creació d'un alumne associat a un grup:

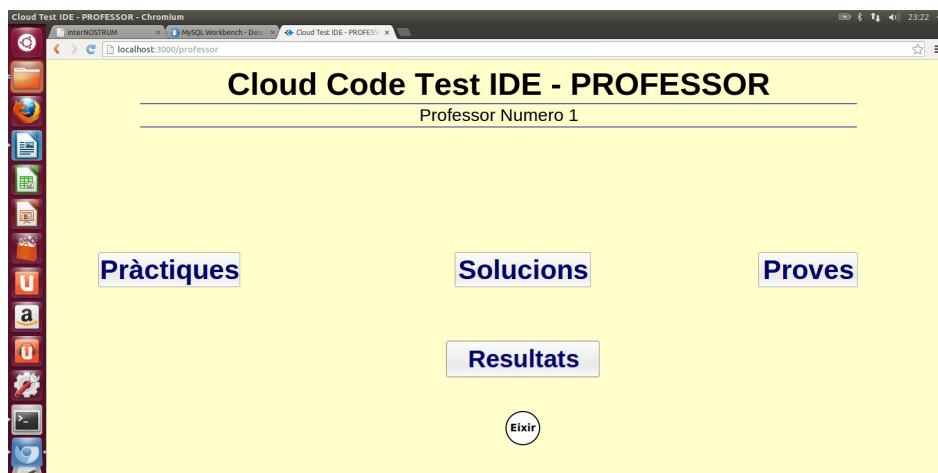


## Professor

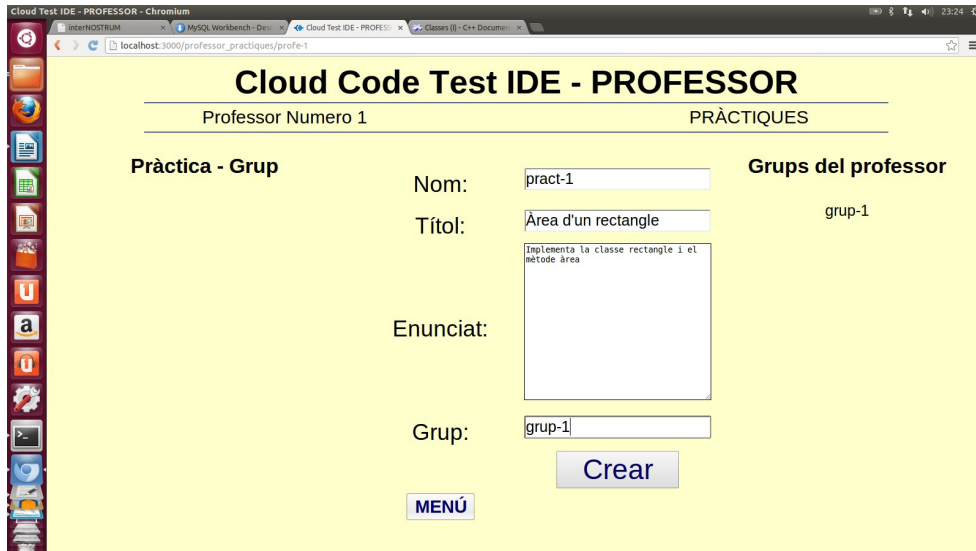
- Entrada al sistema:



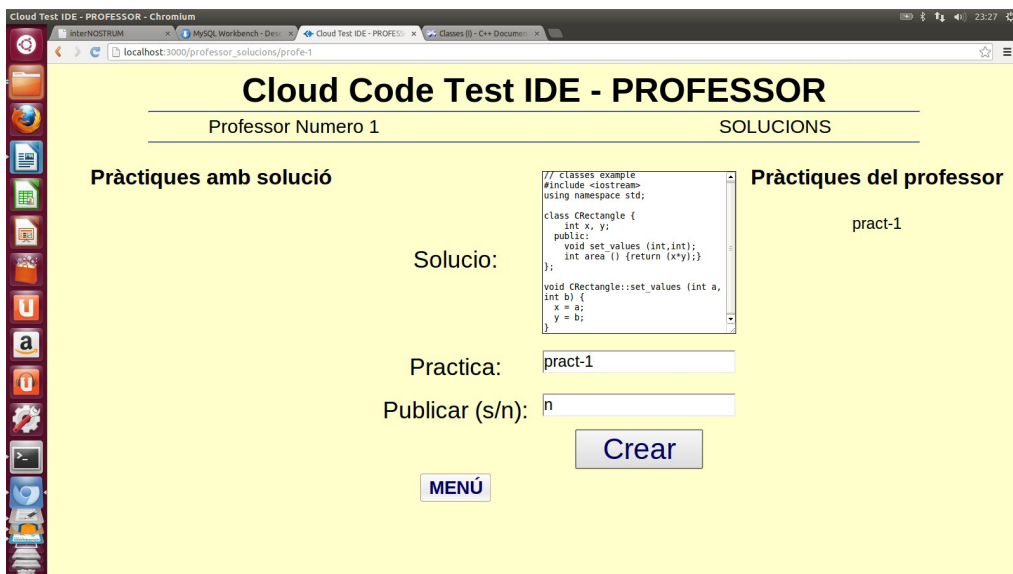
- Apareixen les tasques que pot fer qualsevol professor:



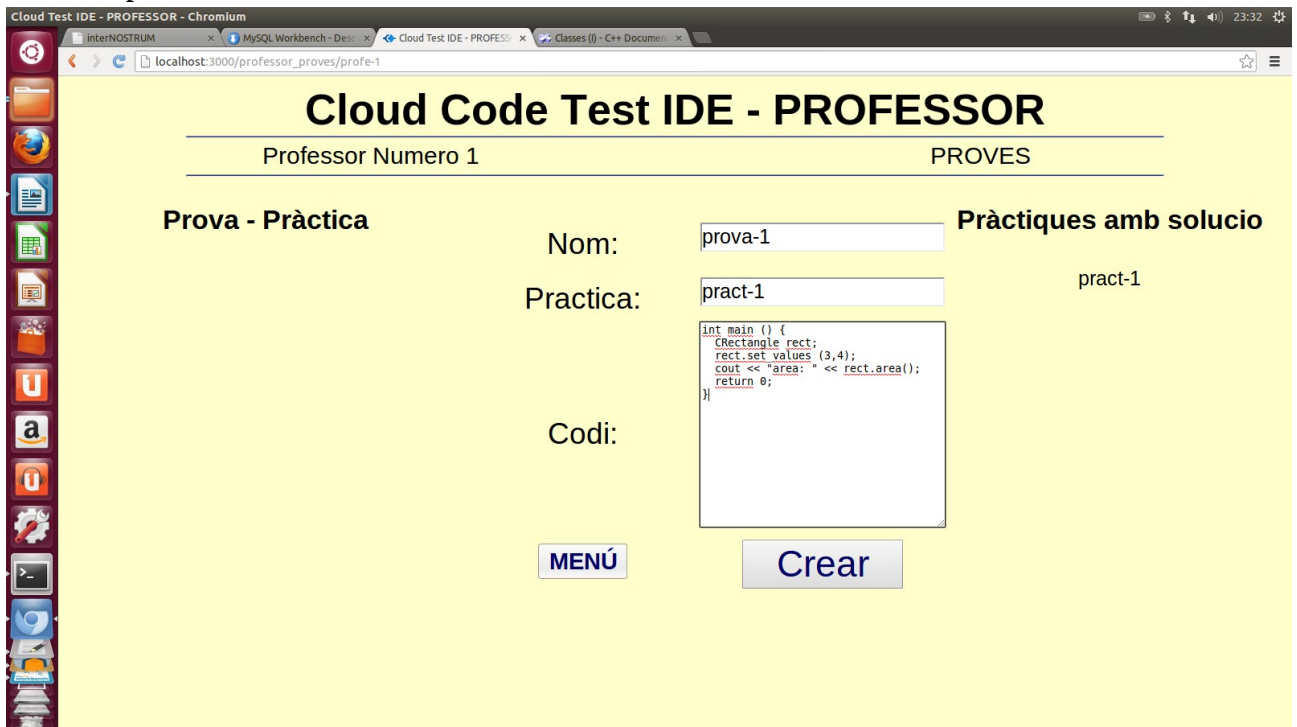
- Entre aquestes el professor pot crear una pràctica:



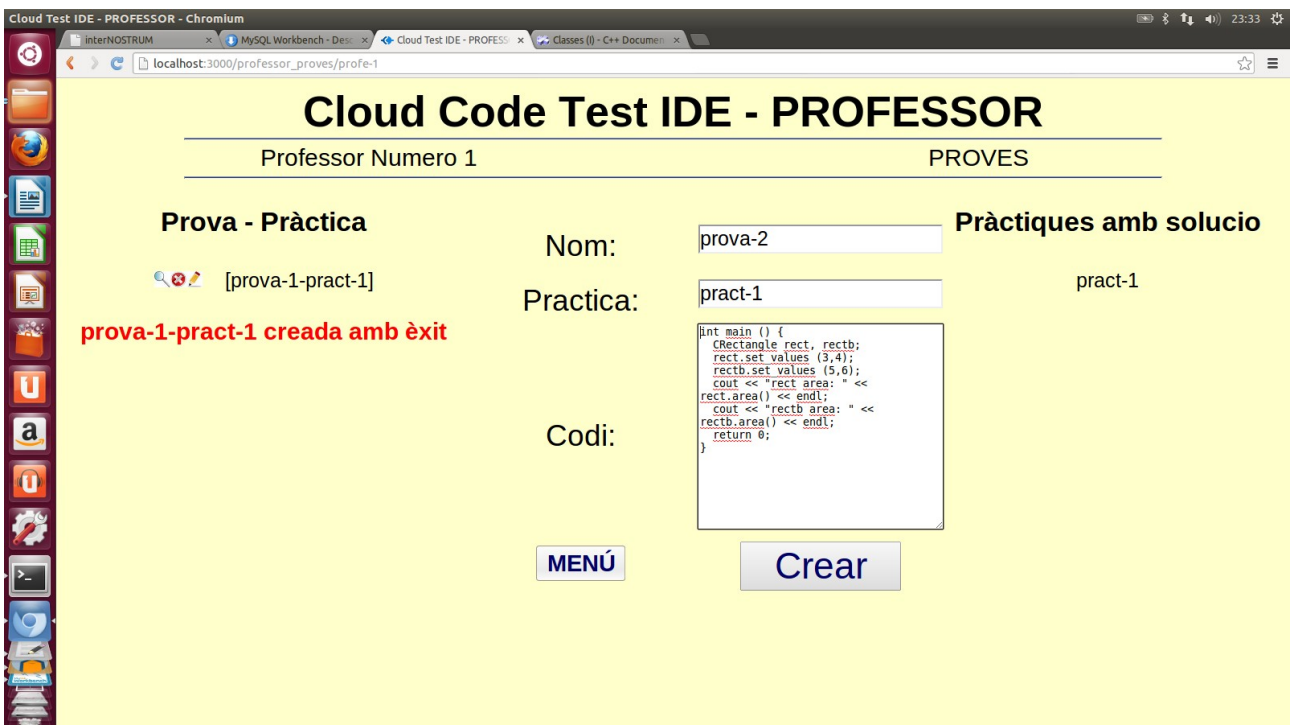
- Pot assignar-li a eixa pràctica una solució:



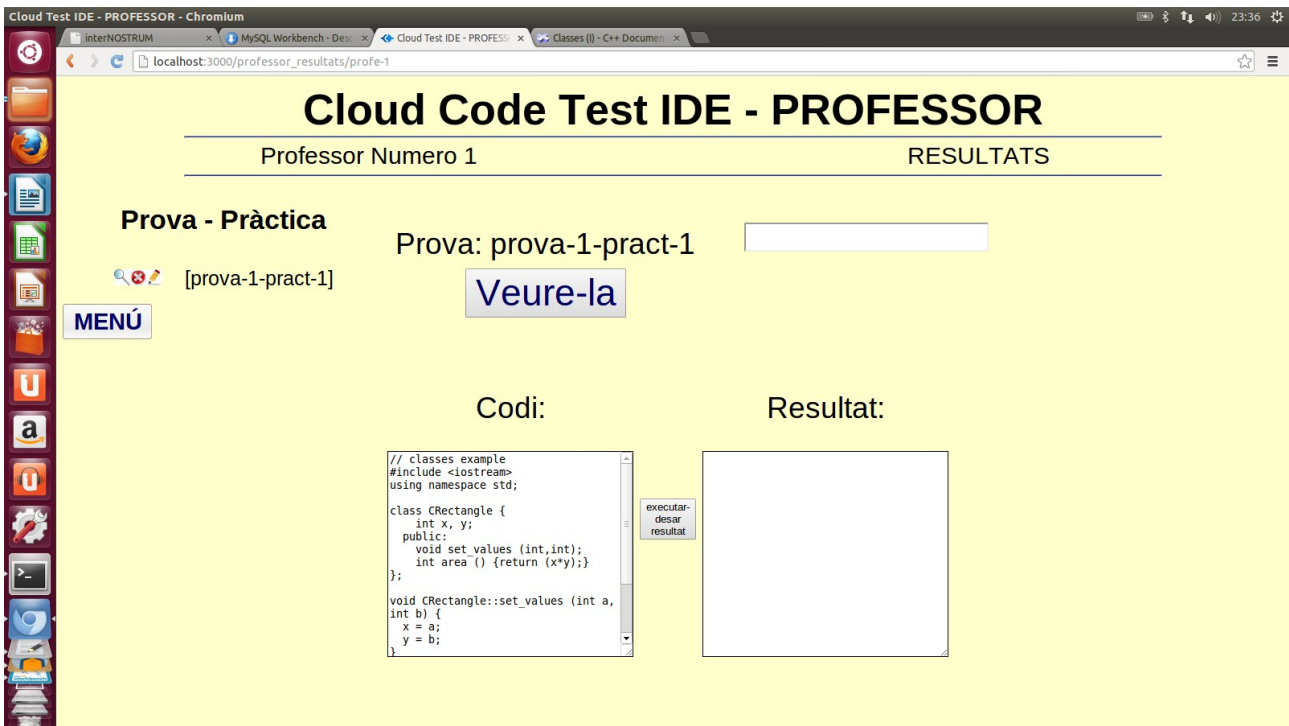
- Per a que l'alumne comprove el seu codi, el professor dona d'alta les proves que es faran a eixes pràctica:



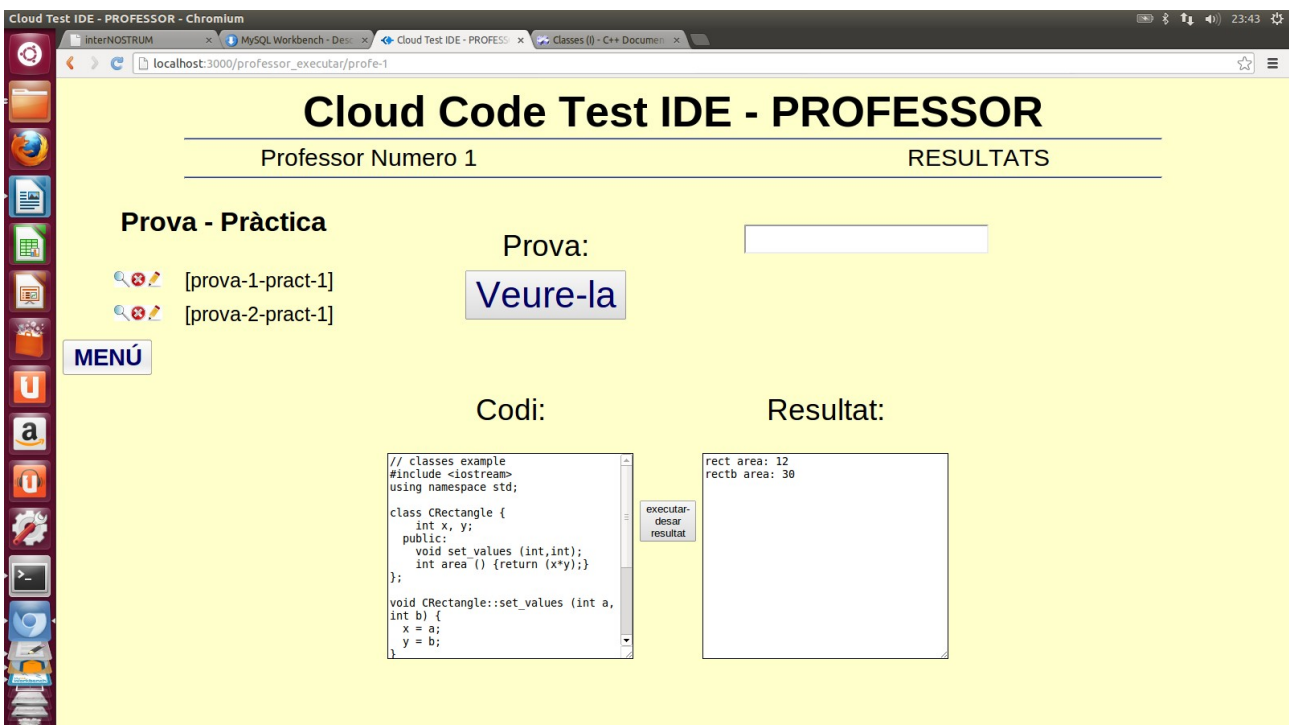
- Si vol, li pot tornar a crear una altra prova:



- I com no, per a cada prova, haurà de crear els resultats esperats per part del codi que implemente l'alumne:



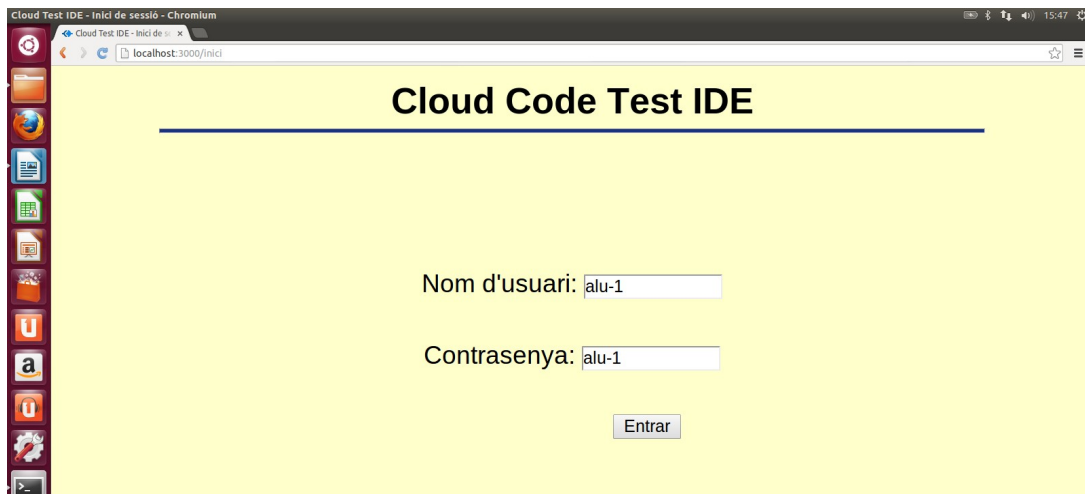
- Al fer un clic al botó Executar – Desar resultat, es visualitzarà el resultat que caldrà obtenir l'alumne i s'emmagatzemarà a la taula corresponent per a que l'alumne comprovi si el resultat que obté és el mateix que l'esperat:



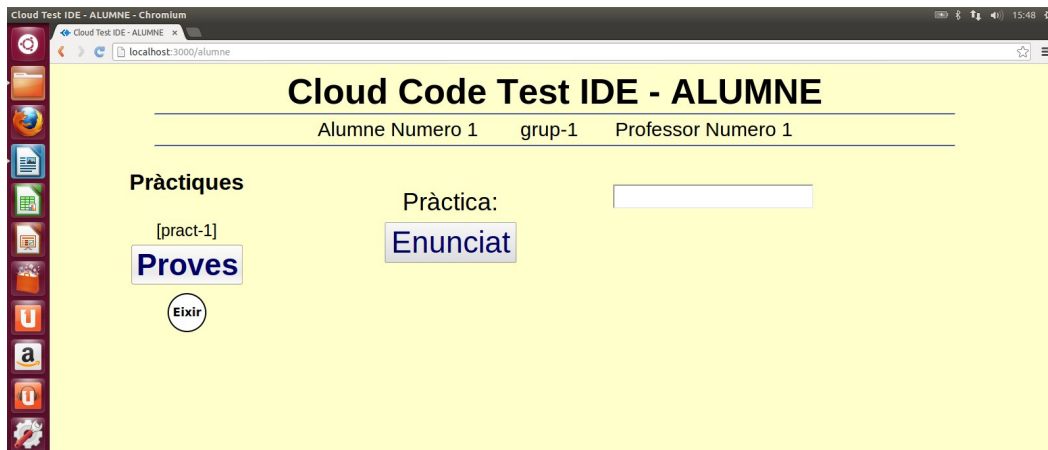
- El mateix farà el professor per a l'altra prova.

## Alumne

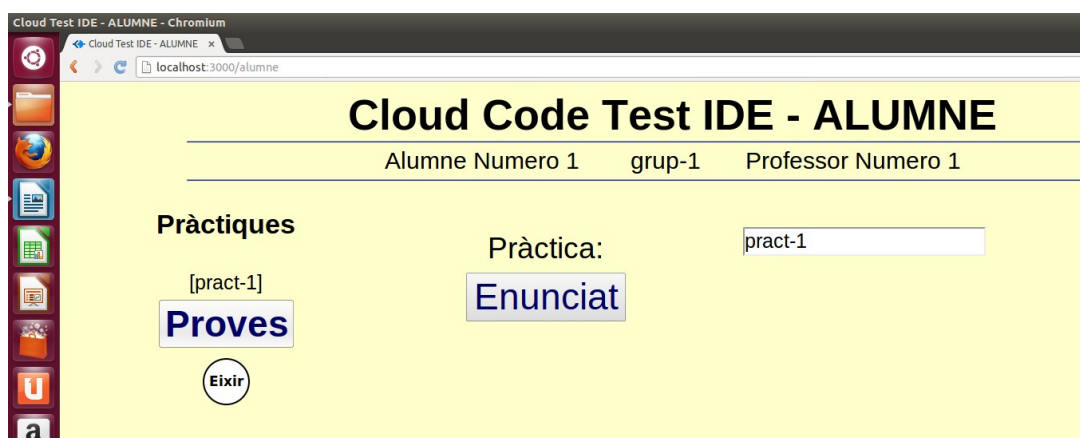
- Entrada al sistema:

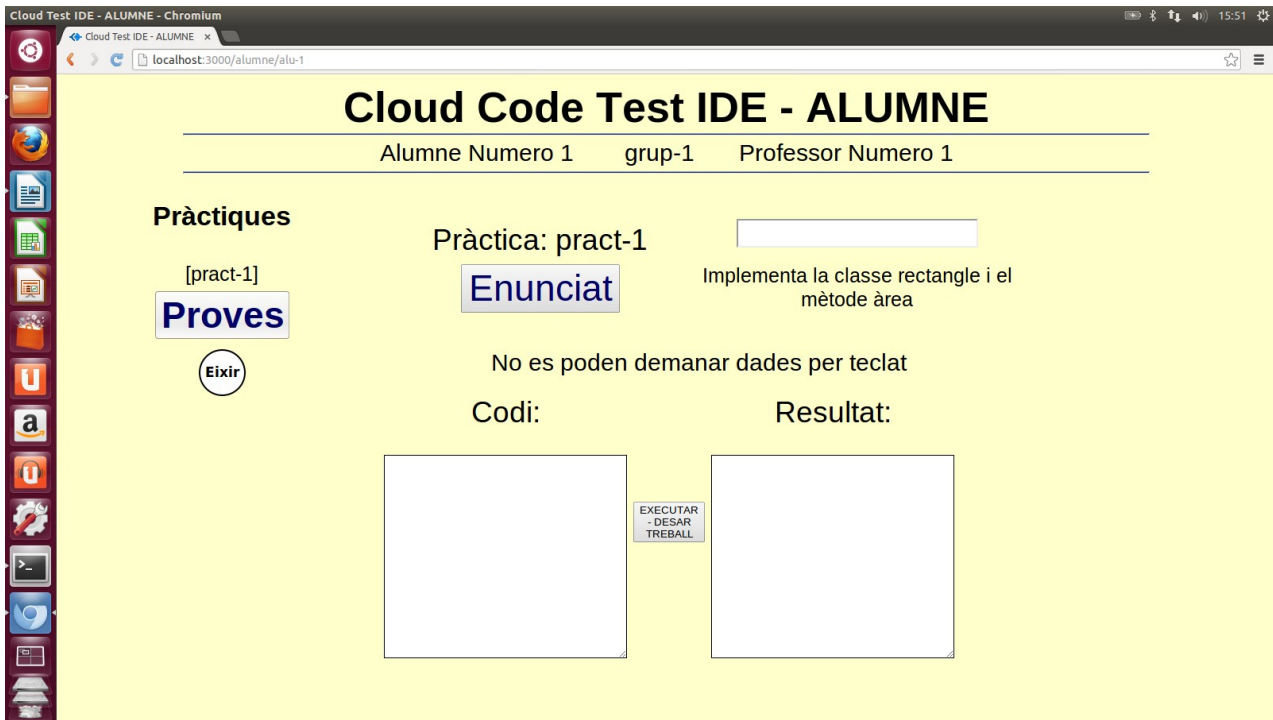


- L'alumne visualitzarà les pràctiques del seu grup:

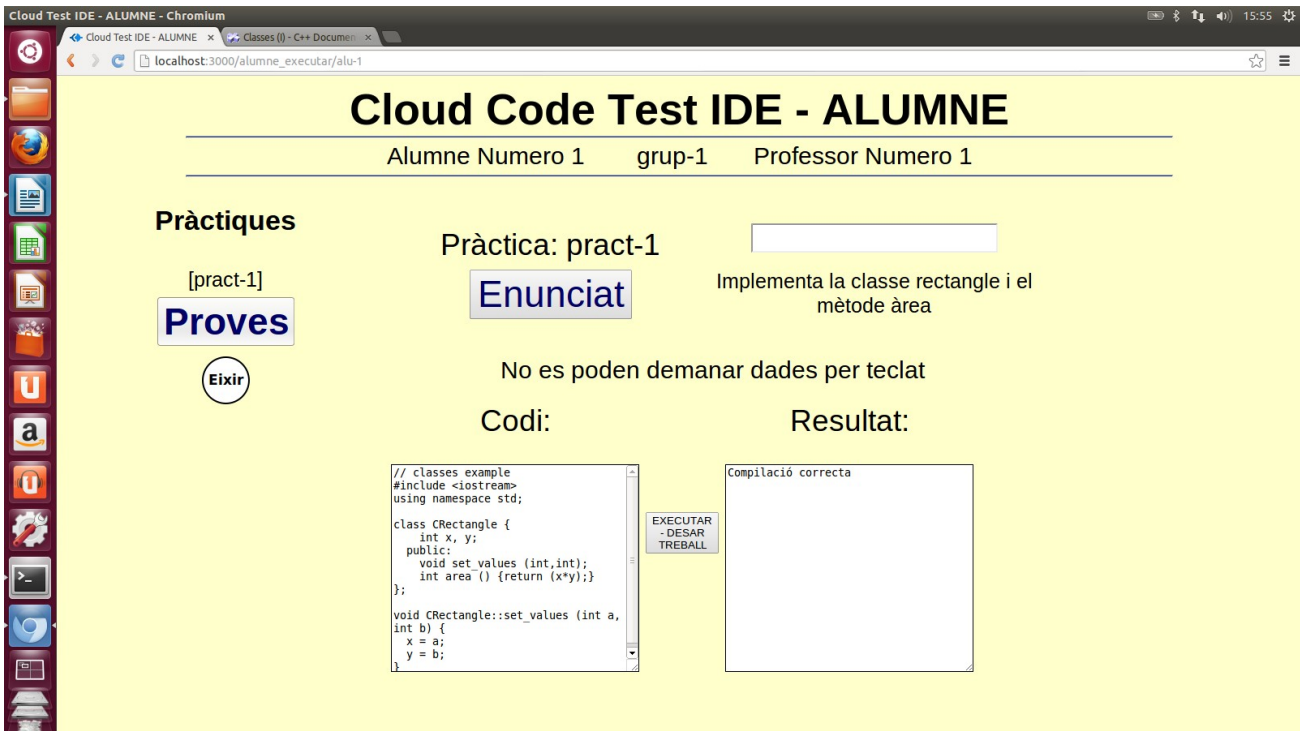


- Indicarà la pràctica amb la qual vol treballar per veure el seu Enunciat:



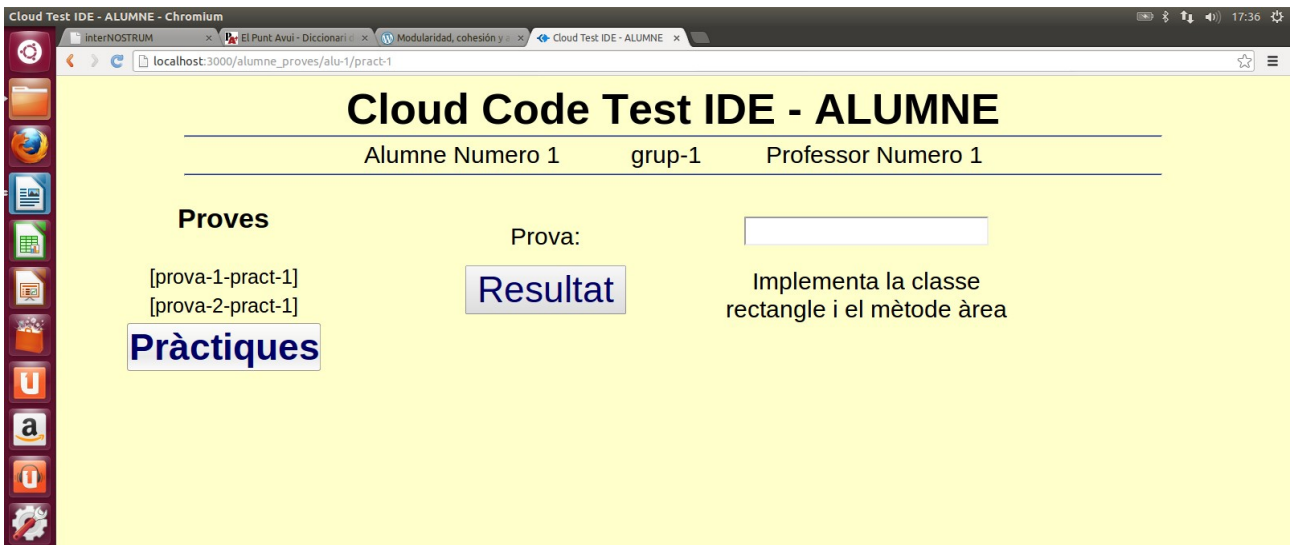


- L'alumne introduirà la seua pràctica, la qual es compila i executa si incorpora la funció main i a més, es desa a la taula corresponent. Aquest codi que escriu l'alumne es desa tan si té errors com si no. Així podrà connectar-se un altre dia per a continuar treballant amb ella:

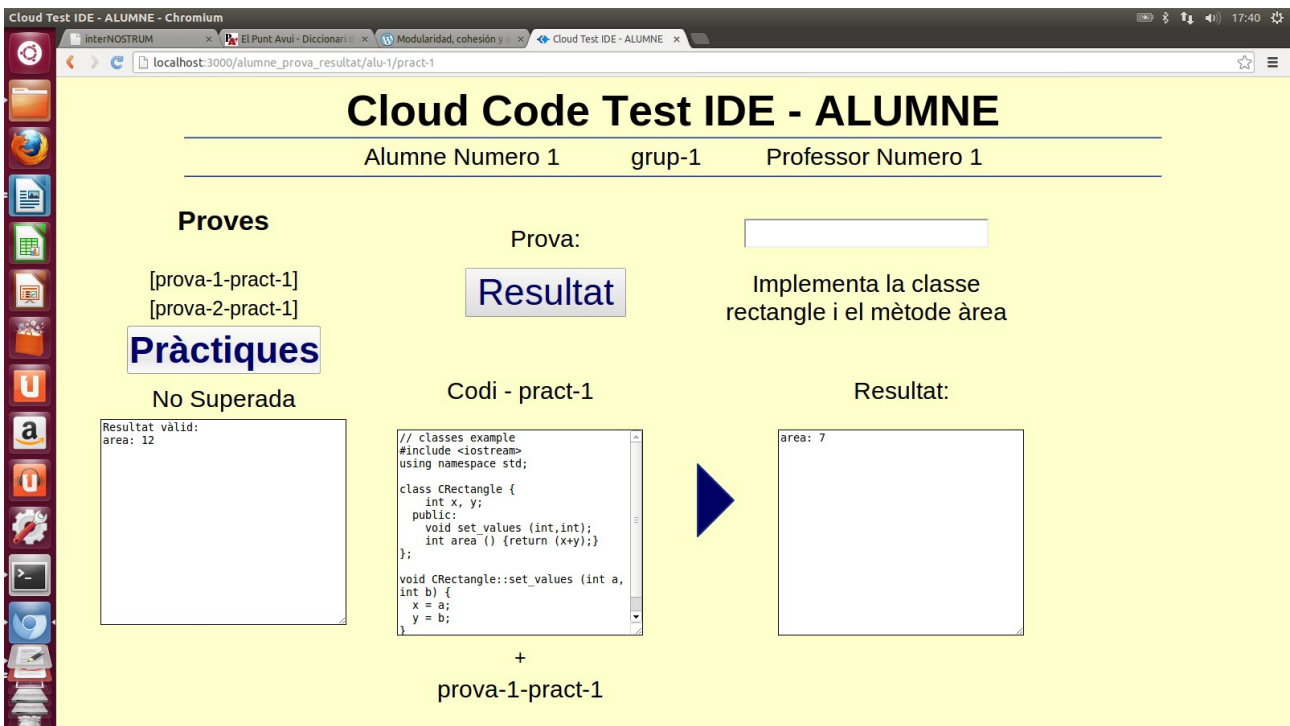


En aquest cas sols a creat la classe i el mètode que es demanava

- Una volta que l'alumne ha compilat la implementació que se li demanava, passarà a veure si passa les proves que li ha proposat el professor. En aquest cas veurà les proves que hi ha per a eixa pràctica i l'enunciat de la pràctica:

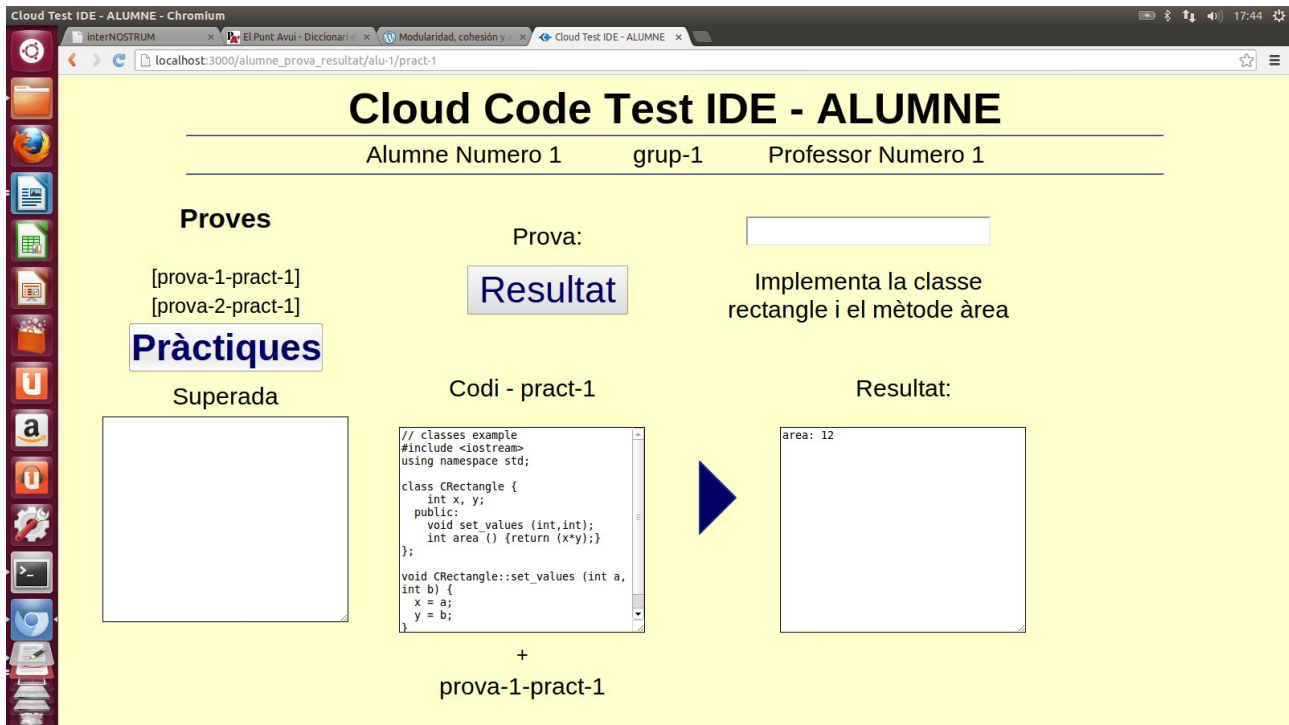


- L'alumne introdueix el nom de la prova que vol comprovar per a la seua pràctica i veu el seu resultat:



En aquest cas l'alumne ha compilat correctament la pràctica però ha obtingut uns resultats que no són els esperats, i per tant no ha superat la prova proposada per l'alumne

- L'alumne torna a pràctiques, comprova l'error comés i el rectifica. Torna a compilar-la (es desa automàticament) i a continuació va a proves per veure si supera la prova.



#### 4.2.4.- Implementació de la lògica

Aquesta aplicació web ha estat implementada amb el framework Express, utilitzant nodejs com a intèrpret de Javascript del costat del servidor. Resumint un poc el seu funcionament, aquest consisteix en anar rebent i dirigint cadascuna de les peticions que sol·liciten els actors i enviant-los (renderitzant) les respostes com a documents Jade que contenen la informació a mostrar, les qual són traduïdes a pàgines web HTML. A banda de les dades obtingudes a la base de dades, molta informació necessària per a poder executar les accions que demanava cada actor és passada mitjançant les pàgines creades amb Jade a les funcions que implementen els gets i posts de les rutes de l'aplicació web. En tot cas, es poden veure comentades a l'arxiu app.js.

A la fase “Desenvolupament Inicial de l'aplicació web” es descriu cadascuna de les carpetes que formen part del lloc web creat amb la combinació de nodejs + express + jade.

Altres aspectes importants a comentar són que conforme s'anava implementant les funcions de cada actor, es feia patent el mal disseny realitzat en la construcció de les entitats – relacions a l'script sql generat amb l'eina Mysql Workbench. El millor haguera sigut tornar a replantejar el disseny inicial (camps, pk, fk, relacions, ...), però com es volia tenir en funcionament les funcions bàsiques de l'IDE al núvol abans de la data de termini, s'ha considerat adaptar, en alguns aspectes, l'aplicació web al disseny inicial. Per exemple, que l'identificador o clau primària de la taula



PROVES és la concatenació de l'identificador de la prova més l'identificador de la pràctica, o. Altre exemple és deixar en blanc el camp nom de la taula ACTIVITAT. Podem considera-ho com si ens hagueren demanat fer aquesta aplicació a partir d'una base de dades existent a l'institut, on es troben totes les dades amb les quals treballen els alumne i professors que imparteixen el llenguatge de programació C++ al centre de secundària. El motiu potser que la creació d'una nova base de dades comportaria una importació i adequació de les dades actuals al nou disseny, i com no, això comportaria unes despeses per al centre que, actualment no s'ho poden permetre.

## 5.- LIMITACIONS I PROVES

### 5.1.- PROVES

Aquesta fase ens ha servit per verificar el funcionament de l'aplicació web per a cadascuna de les funcionalitats associades a cada usuari. Aquestes han consistit en comprovar les següents accions:

- Verificar que un administrador pot crear professors, grups i alumnes. També que pot crear les associacions següents: un grup té associat un professor i un professor pot tenir més d'un grup, un grup pot tenir més d'un alumne i un alumne sols pot estar en un grup.
- Verificar que un professor pot crear una pràctica i associar-la a un o diferents grups que té associat, així com proposar les seues solucions (una per pràctica) i presentar els resultats que un alumne ha d'obtenir per a cada prova que ha proposat el professor per a cada pràctica.
- Verificar que un alumne pot visualitzar les pràctiques associades al seu grup, les quals han estat proposades pel professor vinculat a eixe grup. També s'ha comprovat que un alumne pot escriure el codi C++ per a un enunciat de pràctica i el pot compilar i executar, i que aquesta acció deriva en l'emmagatzematge d'eixa pràctica.

Hem de dir que totes aquestes proves (sobretot a les fases 8 i 9), a part de verificar el correcte funcionament, també ens ha servit per a comprovar el funcionament del codi escrit utilitzant el framework Express i node.js. Al principi del desenvolupament ha costat molt entendre la forma en que s'hauria de programar la lògica de negoci. Per tant, aquestes proves també han servit per autoformar-se en la utilització d'aquestes tecnologies i així poder continuar amb el desenvolupament de l'aplicació.

Però no totes les proves realitzades s'han dedicat exclusivament per a comprovar el correcte funcionament de l'aplicació web. Comprovació de les eines de compilació a LLVM i a Javascript, instal·lació de les tecnologies usades per a implementar l'aplicació, disseny de la base de dades amb l'eina Mysql Workbench, comprovació del funcionament de les plantilles Jade per a renderitzar i passar informació entre unes i altres, repàs d'utilització de les sentències utilitzades en la construcció de la base de dades i les posteriors accions que en ella es poden realitzar (insert, update, ... encara que una de les funcions que no ha donat temps a implementar són les d'esborrat) i rectificació dels anàlisis obtinguts a les fases 6 i 7, han suposat moltes proves per a poder entendre el funcionament i la lògica que haurien d'aplicar en la construcció de l'aplicació.

Altres de les proves que també cal remarcar són el de les pròpies entregues de les PAC. Els comentaris i suggeriments del consultor també han contribuït a comprovar si la línia que s'estava

seguint era la correcta o bé s'hauria de donar un altre enfocament.

## 6.- CONCLUSIONS

### 6.1.- RECAPITULACIÓ

Durant tot aquest procés d'estudi, anàlisi i construcció de l'aplicació web, sempre s'ha tingut en compte els objectius marcats inicialment al pla del projecte. És evident que alguns d'ells no s'han assolit, com per exemple el de donar una major interacció entre els alumnes i els alumnes amb el professorat per a resoldre problemes de programació en C++ mitjançant un entorn web en el núvol. Però per contra, hem obtingut un producte que ens facilita un accés via web a un compilador en C++, i un objectiu que inicialment no vam estipular al pla de treball: adquirir una formació en les tecnologies utilitzades.

Hem vist com no té perquè haver-hi obstacles a l'hora d'executar qualsevol aplicació a la web. Independentment del llenguatge utilitzat en la seua programació, la traducció a un llenguatge intermedi i la seua posterior traducció a un llenguatge propi de la web, ens possibilita la seua execució a la xarxa. LLVM és el pont que ens possibilita la traducció d'una aplicació escrita en qualsevol llenguatge de programació a un llenguatge executable al núvol. Simplement caldrà crear el traductor del llenguatge de programació al codi intermedi LLVM. Al nostre cas, el compilador clang en ha ajudat a realitzar eixa funció, mentre que emscriptem ens ha traduït eixe codi intermedi a un llenguatge propi de la web: Javascript.

En aquest projecte hem vist com la infraestructura LLVM ens ha ajudat a proposar una aplicació didàctica a l'àmbit educatiu. Però aquest pont de connexió que ens dona LLVM també el podem aprofitar per a altres àmbits. Per exemple, si emscriptem ens tradueix codi intermedi LLVM a un llenguatge que es pot executar a la web, un fabricant de plaques base podria construir un traductor de LLVM a instruccions pròpies de la CPU. D'aquesta manera, els fabricants de controladores (targetes de so, targetes de vídeo, ...) simplement crearien un programari que traduiria el *driver* d'eixes controladores a codi LLVM. D'aquesta manera estan fabricant maquinari compatible amb tot aquell maquinari que incorpore el traductor d'LLVM a al seu llenguatge específic. Tot això indica que els àmbits d'aplicació poden ser molt amplis.

També l'elecció de la tecnologia emprada ha estat determinada pels objectius marcats inicials. L'elecció d'una arquitectura web client – servidor del costat del servidor ens ajuda a poder aconseguir les fites establertes al principi.

Per contra, i per a poder assolir la majoria dels objectius, cal desenvolupar un canal xat entre els alumnes i alumnes amb el professor per a que haja més interacció i es pugui donar cobertura a les

diferents modalitats d'ensenyament que ens ofereix el sistema educatiu actual. A part d'això, també s'haurien d'implementar les funcionalitats d'esborrar, edició i visualització que apareixen a les pàgines on treballen els diferents usuaris. La falta de temps, i una major dedicació a funcions que s'han considerat més bàsiques perquè agrupen a un major nombre d'objectius del projecte, no han fet possible desenvolupar-les com caldria.

## 6.2.- AMPLIACIONS

A banda de les funcions no implementades que s'han comentat a l'apartat anterior, algunes de les possibles funcionalitats que podria donar-li un valor afegit per a facilitar la tasca tan del professorat com de l'alumnat poden ser:

### **ADMINISTRADOR**

- Importació del alumnes matriculats a l'assignatura “Tecnologies de la informació i Comunicació I i II”. Als centres de secundària es gestiona la matriculació de l'alumnat a partir del programa Gescen (gestió de centres), encara que actualment s'ha substituït per la plataforma web ITACA. Tots dos incorporen la funcionalitat de poder importar les dades del alumnes matriculats a una assignatura en un format XML. Per a no tenir que donar d'alta un a un els alumnes que pertanyen a un grup, es podria implementar aquesta funció que agafaria aquells camps importants de cada alumnes per a donar-los d'alta a l'IDE al núvol. El mateix es podria amb les dades del professorat.

### **PROFESSORAT**

- Xat on poder resoldre els dubtes d'alumnes que estudien a la modalitat a distància.
- Portar un registre de les voltes que un alumne ha comprovat una prova fins que l'ha resol.
- Un panell d'avaluació on l'alumne pugui veure les qualificacions obtingudes a cada pràctica i un petit comentari fent referència al que ha fet.
- Espai on penjar els continguts de cada tema per a que un alumne els pugui consultar durant la realització de les pràctiques.

### **ALUMNAT**

- Un fòrum on l'alumne pugui obrir un tema de discussió per a resoldre una pràctica i sol·licitar qualsevol tipus d'orientació davant al resultat obtingut en una prova, problemes en la

compilació o per qualsevol altra consulta relacionada amb el continguts a posar en pràctica.

- Espai on els alumnes puguen proposar proves amb els seus resultats.

### 6.3.- VALORACIÓ PERSONAL

Des de que vaig finalitzar la tècnica d'informàtica, quasi tota la meua vida laboral ha estat relacionada amb l'educació en secundària. Per aquest motiu no em veia capaç d'elegir un tema relacionat amb l'àrea de compiladors. He de dir que l'elecció com a primera opció d'aquesta àrea ha estat més influenciada per continuar amb els continguts vistos a les dues assignatures que vaig cursar als últims dos semestres: compiladors I i II. Per això vaig preferir presentar-me i que el consultor em fera la seua proposta.

La lectura de la proposta del projecte em va estimular dues sensacions contràries. Per una banda, prou content de poder fer un projecte relacionat amb la meua tasca docent a l'institut, i per altra, preocupació per la tecnologia a utilitzar per a desenvolupar l'aplicació final del projecte. El desconeixement d'aquesta suposaria un esforç que hauria d'afegir al propi que suposa l'elaboració d'aquest. En tot cas, la primer d'elles ha suposat una gran motivació per poder fer un estudi i anàlisi més exhaustiu de les tecnologies a utilitzar, al qual s'ha dedicat més temps del previst inicialment. Tot i això, durant el procés del projecte anava dissipant-se eixe temor per la utilització de node.js, express i jade. La principal i única font d'informació ha estat Internet, on he tingut que consultar molts dels recursos usats, i amb diferents formats, per a poder posar en pràctica aquestes tecnologies. Conforme avançava amb la construcció de l'aplicació, se me'n feia més fàcil la utilització d'aquestes. Per això moltes de les implementacions fetes al principi no segueixen un patró i es troben més embolicades que la resta que vaig implementar al final.

Altre aspecte que destacaria d'aquest projecte és que encara que la proposta va ser feta pel consultor, aquesta l'he feta pròpia, portant-la a l'àmbit on jo desenvolupe la meua tasca professional. L'elaboració del projecte ha tingut en compte tot el material que usualment hi ha en un aula d'informàtica a qualsevol centre de secundària del País Valencià, tenint present els avantatges i desavantatges d'aquest. Des del principi, el consultor em va orientar que la millor forma de poder elaborar un projecte és portar-se'l a un terreny particular. D'aquesta forma es treballaria d'una forma més amena, gaudint dels resultats obtinguts durant tot el procés del projecte. La valoració que faig d'aquest punt és prou positiu. A més, ha estat la base motivadora per anar elaborant cadascuna del les fases del projecte.

Respecte a l'aplicació final obtinguda, aquesta proporciona les funcionalitats bàsiques que es requereixen en un entorn IDE al núvol per a poder practicar amb el llenguatge C++, tot i que

algunes que apareixen a les interfícies no han estat implementades (eliminar, editar i visualitzar). En canvi, les seues interfícies no són les dissenyades inicialment. Un estudi de JQuery, haguera suposat una temps afegit que no podia gastar i per tant vaig decidir per implementar interfícies adequades als continguts que fins eixos moments havia estudiat i practicat amb Jade. Per poder comprovar les seues utilitats a l'àmbit educatiu, dos alumnes de segon de batxillerat van provar l'aplicació. Tots dos no van tenir dificultat d'entendre el seu funcionament, però en canvi, si van veure que l'espai destinat a la programació era prou petit i calia una espai molt més ampli.

Per últim, he de dir que la valoració personal que faig de tot el conjunt del projecte és molt positiva. L'aplicació de continguts vistos a altres assignatures, el treball realitzat a l'estudi i anàlisi de les tecnologies que he utilitzat per a compilar i executar codi C++ a la web, així com aquelles que m'han servit en la construcció del lloc web, el resultat final de l'aplicació web i la seua utilitat dins del context educatiu, m'ha servit per complementar la meua formació i deixar oberta altres tipus d'aplicacions que pogueren ajudar al docent d'informàtica en un centre de secundària.

## 7.- REFERÈNCIES

### Arquitectura

[http://www.ehowenespanol.com/diferencia-lado-del-cliente-programacion-del-lado-del-servidor-sobre\\_151711/](http://www.ehowenespanol.com/diferencia-lado-del-cliente-programacion-del-lado-del-servidor-sobre_151711/)

<http://es.wikipedia.org/wiki/Cliente-servidor>

### Emscripten

<https://github.com/kripken/emscripten/wiki>

<https://github.com/kripken/emscripten/wiki/Tutorial>

### LLVM

<http://llvm.org/>

<http://www.aosabook.org/en/llvm.html>

<http://es.softuses.com/108664>

<http://www.meneame.net/story/comparativa-rendimiento-entre-clang-gcc>

<https://github.com/kripken/emscripten/wiki/Getting-Started-on-Ubuntu-12.10>

### IDE al núvol

<http://www.salesforce.com/es/cloudcomputing/>

<http://knowurpc.wordpress.com/2012/05/07/sourcelair/>

<https://www.citycloud.com.ar/city-cloud-es/usar-servidores-virtuales-para-compilar-aplicaciones/>

[http://es.wikipedia.org/wiki/Computaci%C3%B3n\\_en\\_la\\_nube](http://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube)

<http://unlugarmuygeek.blogspot.com.es/2013/01/sourcelair-un-ide-en-la-nube.html>

<http://blog.jorgeivanmeza.com/2012/03/instalando-el-ide-de-cloud9-en-gnulinix-ubuntu/>

<http://www.bonillaware.com/cloud9>

<https://c9.io/site/features/>

<http://chalchicha.es/entorno-de-desarrollo-en-la-nube-para-javascript-cloud9-github-y-heroku.html>

<http://www.visualbeta.es/34337/aplicaciones-web/cloud9-ide-un-entorno-de-desarrollo-colaborativo-y-en-la-nube/>

<http://tecnodiseno.com/prueba-tus-scripts-de-forma-online-con-ideone/>

<http://www.genbeta.com/herramientas/ideone-tu-compilador-online>

<http://qiezi.me/projects/mruby-web-irb/mruby.html>

### Node.js

<http://nodejs-es.github.io/api/all.html>

<http://nodejs.org/docs/v0.4.10/api/http.html#hTTP>

<http://fernetjs.com/2011/12/configurando-nodejs/>

<http://oodavid.tumblr.com/post/15090798307/how-to-install-node-js-on-linux>

<http://www.nodehispano.com/2011/11/como-instalar-node-js-en-ubuntu-linux-nodejs/>

<https://github.com/joyent/node/wiki/Installing-Node.js-via-package-manager>

<http://www.desarrolloweb.com/articulos/intro-nodejs.html>

<http://es.wikipedia.org/wiki/Nodejs>

<http://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/>

<http://www.joinebook.com/curso-de-node-js/>

<http://fernetjs.com/2012/02/estructura-de-un-sitio-web-mvc-en-nodejs/>

<http://www.genbetadev.com/frameworks/introduccion-a-la-programacion-asincrona-con-nodejs>

<http://www.genbetadev.com/frameworks/introduccion-a-la-programacion-asincrona-con-nodejs>



## ejs-ii

<http://www.nodebeginner.org/index-es.html>

<http://www.aprendiendonodejs.com/2011/10/como-utilizar-npm.html>

<http://embat.es/labs/labs-un-poco-de-node-js-y-socket-io.html>

<https://github.com/hermanjunge/herman-sessions-tutorial-101>

## **Express**

<http://expressjs.com/>

<http://geeks.ms/blogs/dsalgado/archive/2012/03/23/taller-de-nodejs-express-y-azure-en-el-codemotion.aspx>

<http://community.logicalbricks.com/node/181>

<http://blog.koalite.com/2011/11/tutorial-node-js-express-jquery-iv-conclusiones-y-proximos-pasos/>

## **Jade**

<http://jade-lang.com/>

<http://jhernandez.es/noticia/plantillas-nodejs>

<http://blog.koalite.com/2011/11/tutorial-node-js-express-jquery-iv-conclusiones-y-proximos-pasos/>

<http://embat.es/post-tecnico/jade-la-elegancia-y-sencillez-en-el-html.html>

<https://github.com/visionmedia/jade>

<http://www.youtube.com/watch?v=P0Bq3SsQARs>

[https://github.com/alessioalex/NodeTuts/blob/master/express\\_samples/users.js](https://github.com/alessioalex/NodeTuts/blob/master/express_samples/users.js)

<http://scalate.fusesource.org/documentation/jade-syntax.html>

## **Apunts UOC**

Apunts de les assignatures “*Compilador I*” i “*Compilador II*”.

Consulta dels materials UOC de les assignatures “*Enginyeria del programari orientat a l'objecte*”, “*Arquitectura de sistemes distribuïts*” i “*Enginyeria del programari de components i sistemes distribuïts*”.

Consulta dels material UOC de les assignatures “*Interfícies multimèdia*” i “*Interacció humana amb els ordinadors*”.

## **Mysql**

<http://www.youtube.com/watch?v=EklwX0Myr38>

<http://www.desarrolloweb.com/manuales/9/>

<http://stackoverflow.com/questions/5878266/using-mysql-with-nodejs-and-express-node-mysql>

<http://nodejsrocks.blogspot.com.es/2012/04/nodejs-expressjs-mysql.html>

<https://github.com/felixge/node-mysql>

## **C++**

<http://www.cplusplus.com/doc/tutorial/>

## **Terminologia catalana**

[www.termcat.cat](http://www.termcat.cat)