



Introducción a la Programación

Universidad Católica Boliviana

MSc, José Jesús Cabrera Pantoja

Jose Jesus Cabrera Pantoja



- Formación:
 - BSc, Mecatrónica y Robótica. Sistemas Robóticos Distribuidos
 - MSc, Mecatrónica y Robótica. Robots industriales y de servicio
- Experiencia en la Academia:
 - Centro de investigación y educación "Robotics at BMSTU"
 - Investigación en Computer Vision, Control Systems, Robotics
- Experiencia en la Industria:
 - GAZ Group, Industria Automotriz
 - Carlos Caballero SRL

Objetivos del curso

- **Objetivo General:** Producir programas de software para resolver problemas de baja complejidad aplicando el paradigma estructurado.

Objetivos del curso

- **Objetivo General:** Producir programas de software para resolver problemas de baja complejidad aplicando el paradigma estructurado.
- **Resolver problemas**
 - Usar técnicas de programación para la resolución de problemas de baja complejidad mediante un lenguaje de programación
 - C++

Es decir queremos...

- Hacer que las computadoras no entiendan para resolver una serie de problemas sencillos.
- Entender como la computadora ejecuta nuestro código.
- Seguir las buenas practicas al momento de resolver problemas

Hoy veremos

- Introducción a la programación
- Course Overview
- Organización

Introducción a la Programación

¿Que representan estos números?

085 074 193 180 110 092

221 180 041 106 180 123

039 087 084 061 012 093

098 074 102 098 034 111

145 128 255 233 234 124

¿Que representan estos números?



¿Que representan estos números?

0x33 0x10 0x20 0x42 0xAF

0x12 0xAB 0xBE 0xFF 0xAA

0xEA 0xCC 0xDC 0xDC 0xAF

0x49 0x87 0x21 0x22 0xFF

0x9A 0x8F 0xFF 0x32 0xEE

¿Que podemos hacer?

- Escriba sus propuestas

Programa

- La computadora sea capaz de entender nuestras ordenes
- ¿Cómo puede entendernos?
 - Le hablamos
 - Le escribimos una carta
 - Le pasamos un serie de instrucciones escritas en español
 - Le pasamos un serie de instrucciones escritas en ingles
 - No podemos comunicarnos con la computadora
 - ...

Programa

- La computadora sea capaz de entender nuestras ordenes
- Escribimos una serie de instrucciones mediante un lenguaje de programación en nuestro caso C++



```
1 std::cout << "Hello from C++ :D" << std::endl;
```

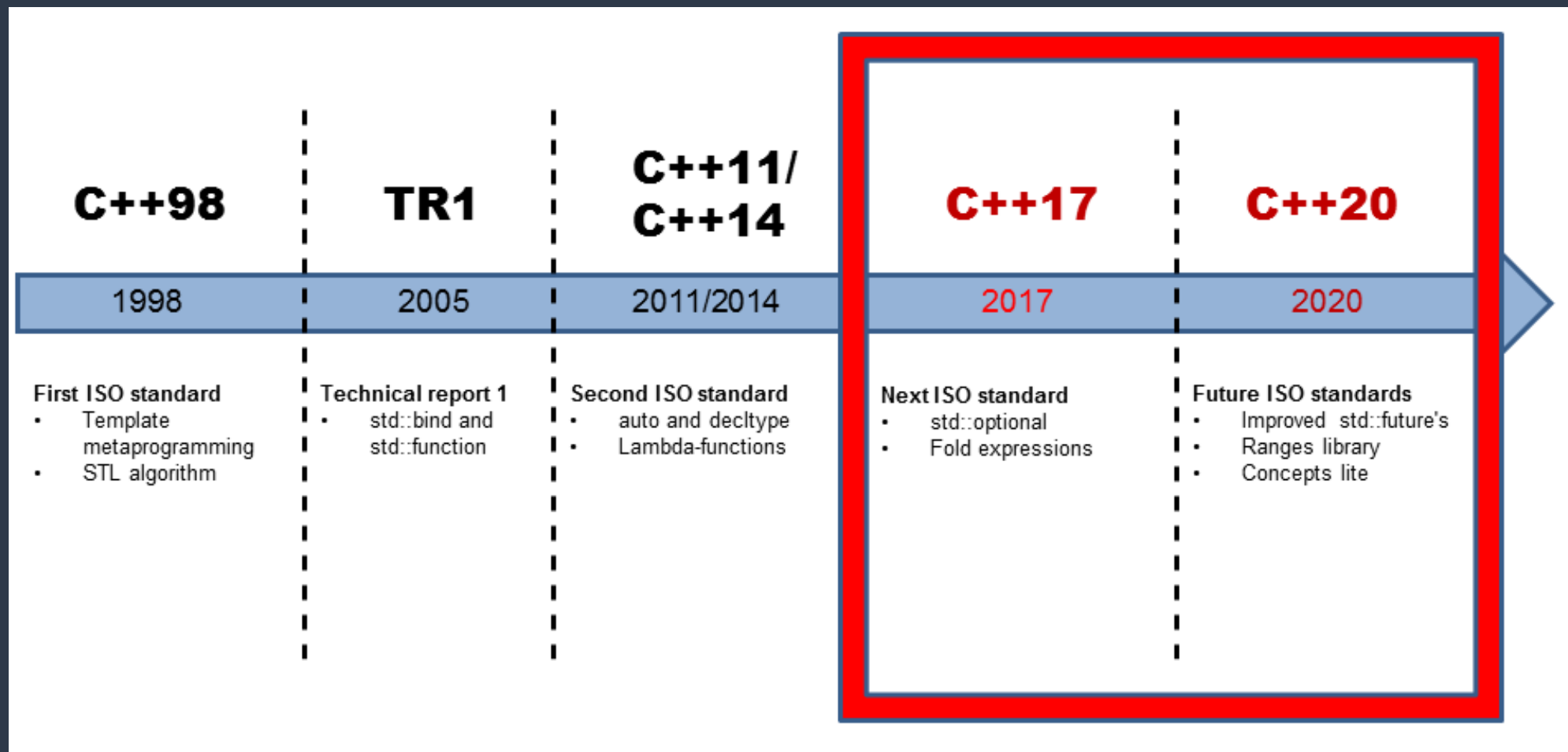
C++

- C++ es un lenguaje complicado
- Rico en características
- Constantemente en evolución y adaptación
- Con una gran comunidad!
- C++ es un lenguaje **compilado**!

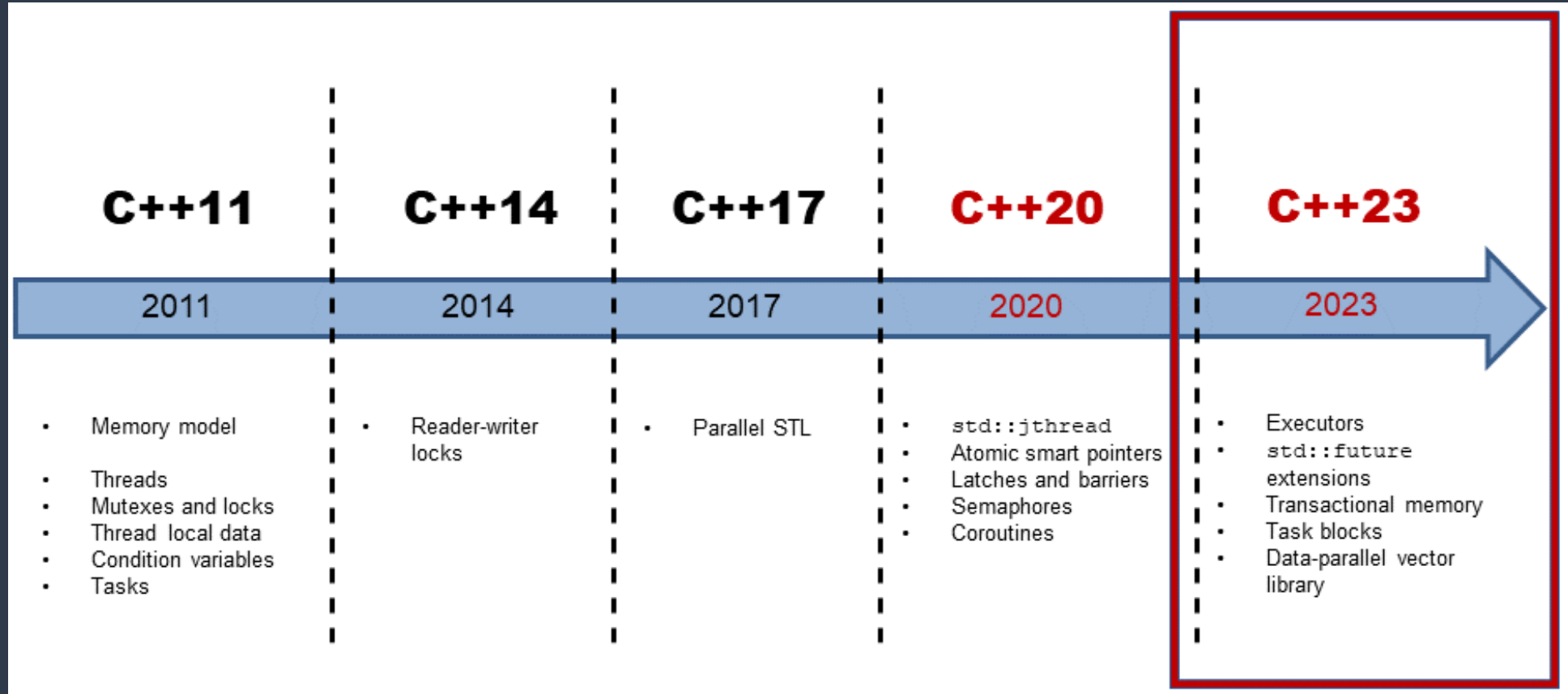


International
Organization for
Standardization

C++



C++





FEATURES

[Current ISO C++ status](#)

[Upcoming ISO C++ meetings](#)

[Upcoming C++ conferences](#)

[Compiler conformance status](#)

STANDARDIZATION

[The Standard](#)

[The Committee](#)

[Meetings and Participation](#)

[Current Status](#)

[How To Submit a New Issue / Defect Report](#)

[How To Submit a Proposal](#)

[The life of an ISO proposal: From "cool idea" to "international standard"](#)

[Library Design Guidelines](#)

[Standing Documents](#)

[ISO/IEC JTC1 Procedures](#)

[Forums](#)

[Sign In / Suggest an Article](#) [Register](#)

[Get Started!](#) [Tour](#) [Core Guidelines](#) [Super-FAQ](#) [Standardization](#) [About](#)

[Home](#) » [Standardization](#) » [The Standard](#)

The Standard

The current ISO C++ standard is officially known as ISO *International Standard ISO/IEC 14882:2020(E) – Programming Language C++*.

Where to get the current standard (C++20)

[Purchase the C++20 official standard](#). You can purchase the official standard at the ISO Store or at national body stores such as the ANSI store.



International
Organization for
Standardization

Where to find related materials (in-progress C++23)

The in-progress LaTeX source materials are [maintained on GitHub](#). Check out the repo to get the current snapshot. Note that this does not correspond to the final text of any published standard, but it can be used for general unofficial reference to answer basic questions about C++.

Also see sites like [cppreference.com](#) and [cplusplus.com](#), which are not authoritative but also can be used for general unofficial reference to answer basic questions about C++.

FAQs

Q: Why is the standard hard to read? I'm having trouble learning C++ from reading it.

The standard is not intended to teach how to use C++. Rather, it is an international treaty – a formal, legal, and sometimes mind-numbingly detailed technical document intended primarily for people writing C++ compilers and standard library implementations.

Fortunately, there are lots of good books that do teach how to use C++! [See these recommendations](#) as a starting point for high-quality tutorial and reference information about how to learn and use C++.

Q: Why are the C++ working materials freely available on GitHub when the standard must be purchased from ISO or another standards organization?

github

PUBLIC



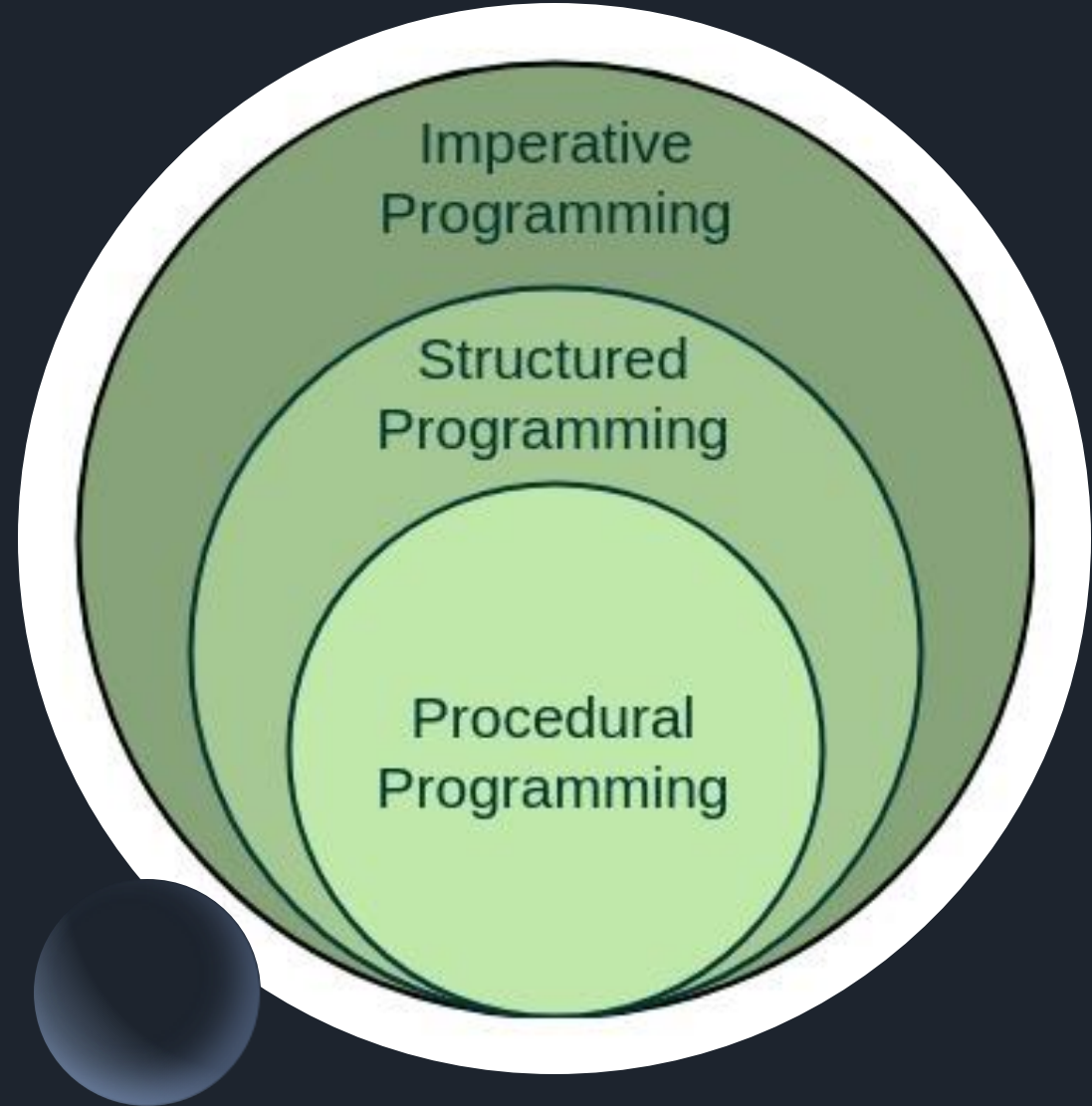
[cplusplus](#) / [draft](#)

Code

C++ standards drafts — [Read](#)

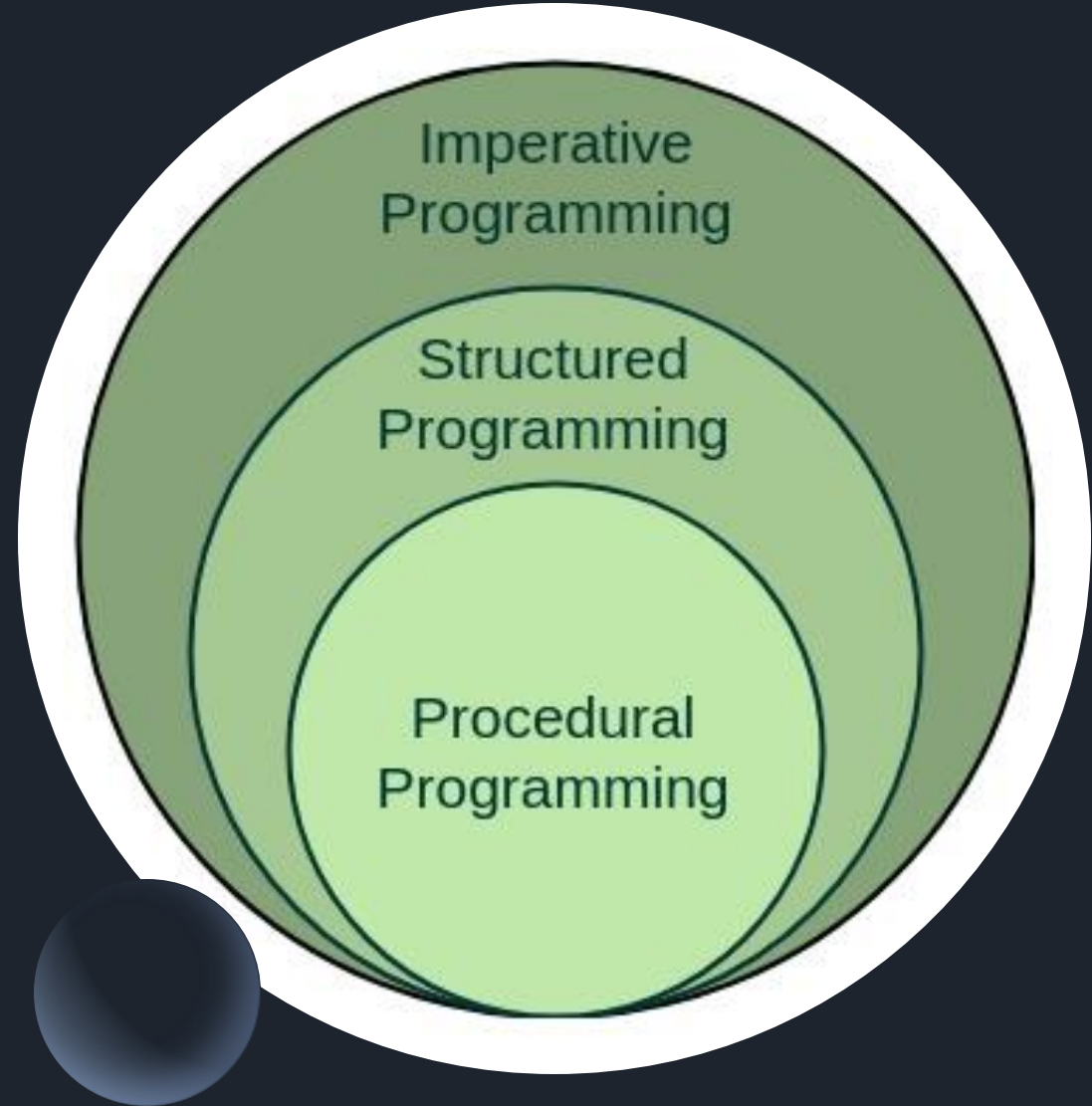
Programas de computadora

- Varias formas de programar:
 - Programación imperativa
 - Programación estructurada
 - Programación procedimental



Programación procedimental

- Secuencia tras secuencia
- Una instrucción a la vez en un orden específico (Una tras otra)

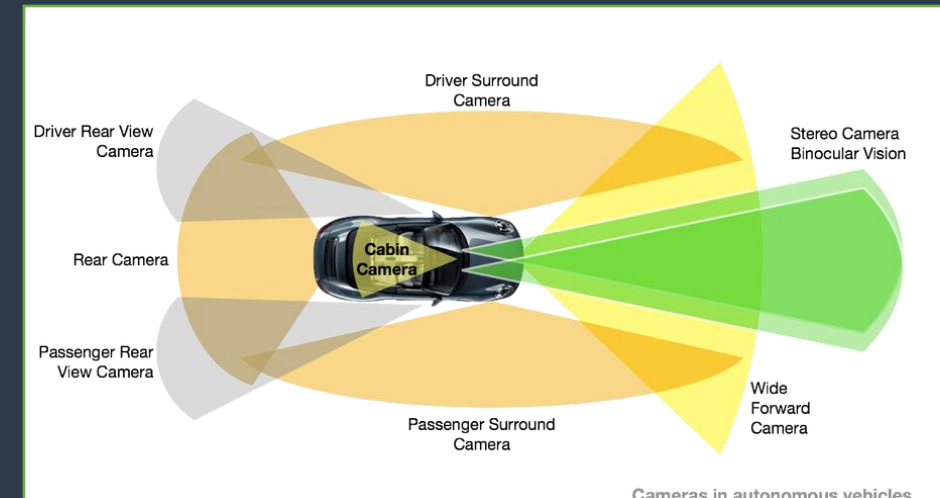
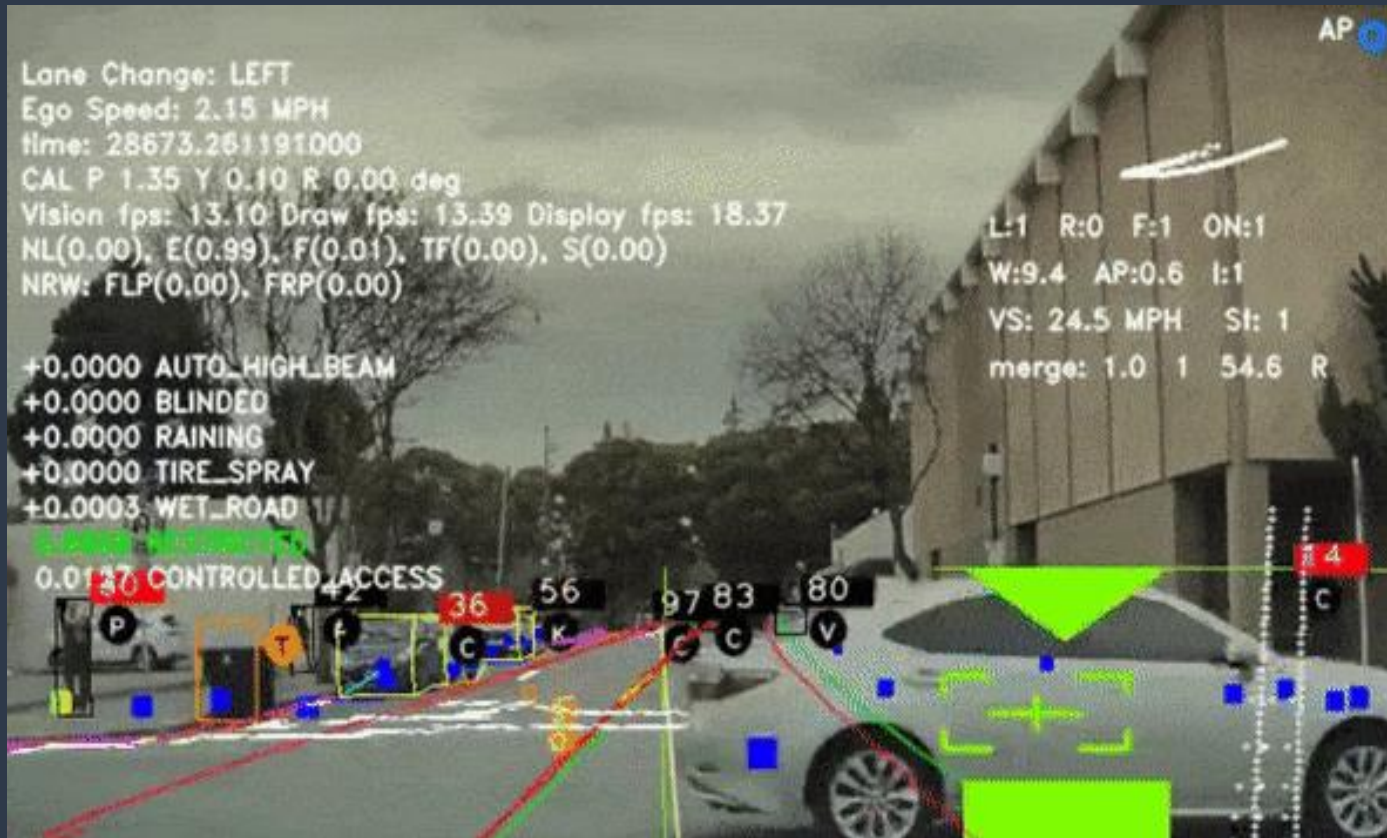


¿Por que C++?



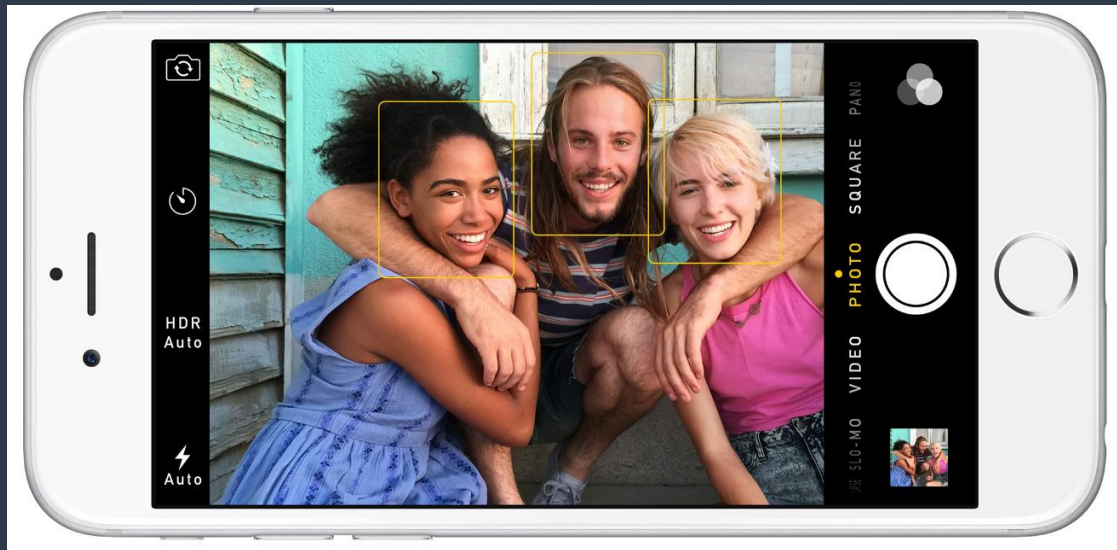
Ejemplos

- Self Driving cars



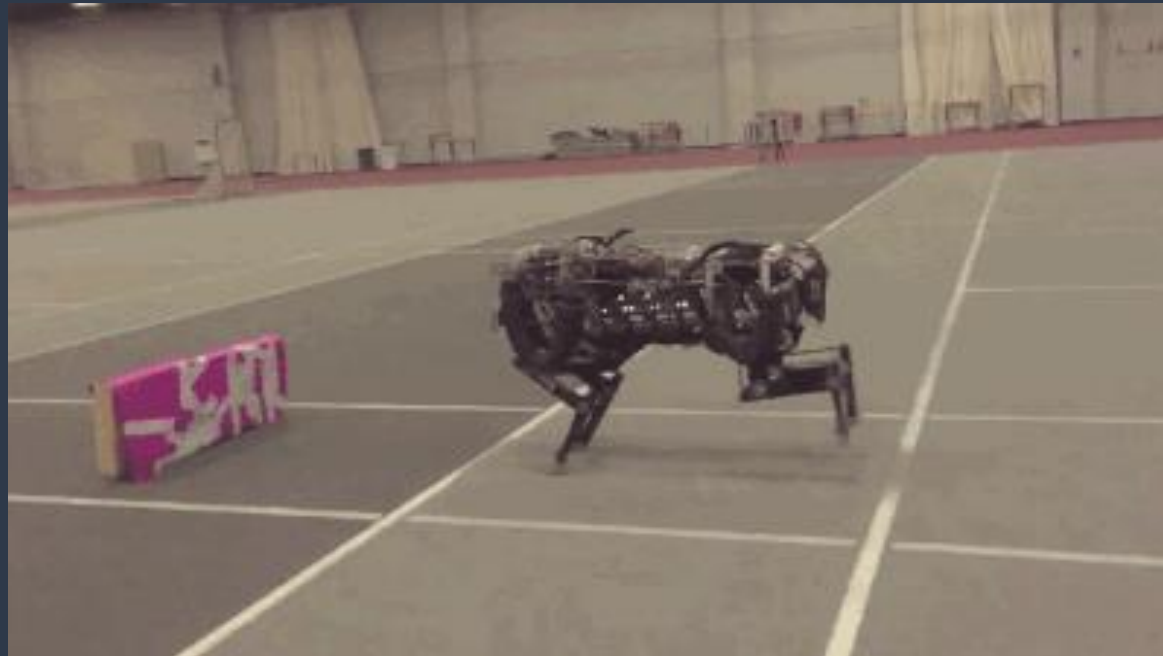
Ejemplos

- Detección de objetos



Ejemplos

- Permite que los robots puedan moverse



Ejemplos

- Algunas empresas también lo usan:

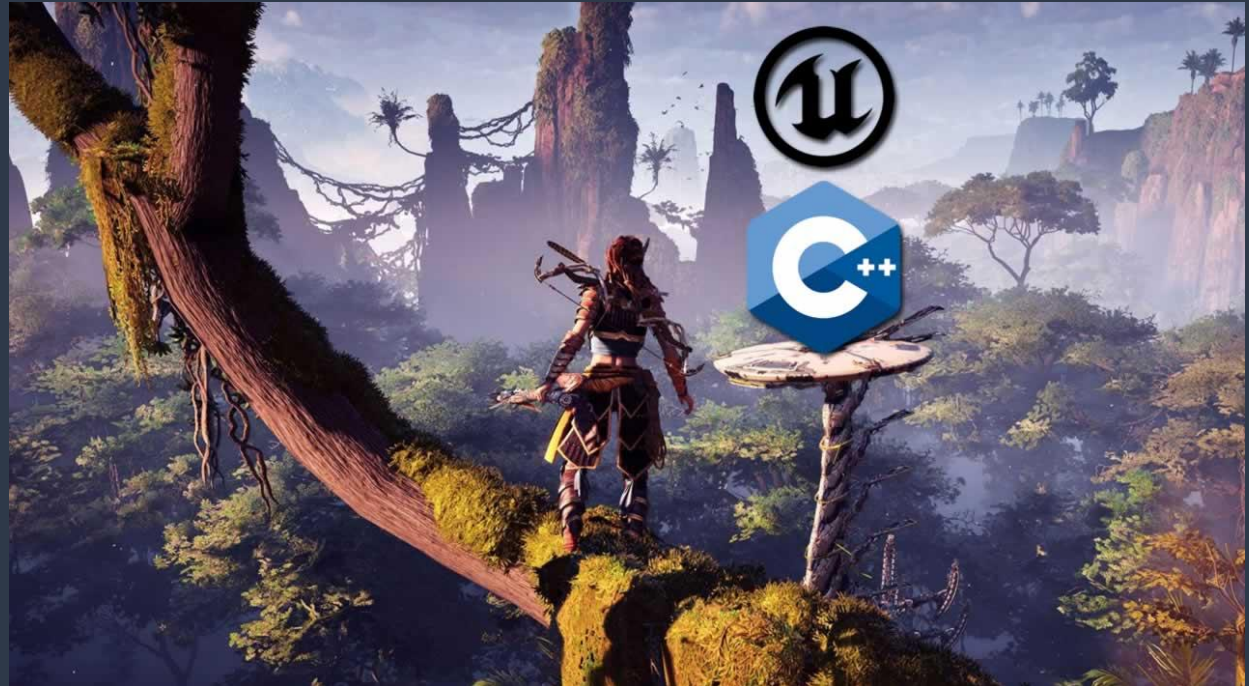


Microsoft

Google

Ejemplos

- Video juegos



C++ resuelve...

- C++ es adecuado para el desarrollo de alto y bajo nivel. Se utiliza para crear pequeños proyectos, grandes productos e incluso sistemas operativos.
- A pesar del adiós regular a C ++, todavía está escrito en Amazon, Microsoft, Google, Adobe.
- Después de todo, este lenguaje aún no tiene análogos en términos de rendimiento.

En resumen ...

- Back-end de alto rendimiento, (Sistema de búsqueda por ejemplo)
- Aplicaciones de red, (Comunicaciones entre redes)
- Desarrollo de juegos,
- Desarrollo de equipos (sistemas embebidos, robótica, drones, dispositivos domésticos inteligentes, IoT).

Dato curioso :D

- Debido a su potencia y facilidad de uso, C y C++ se utilizaron en la programación de los efectos especiales de Star Wars.

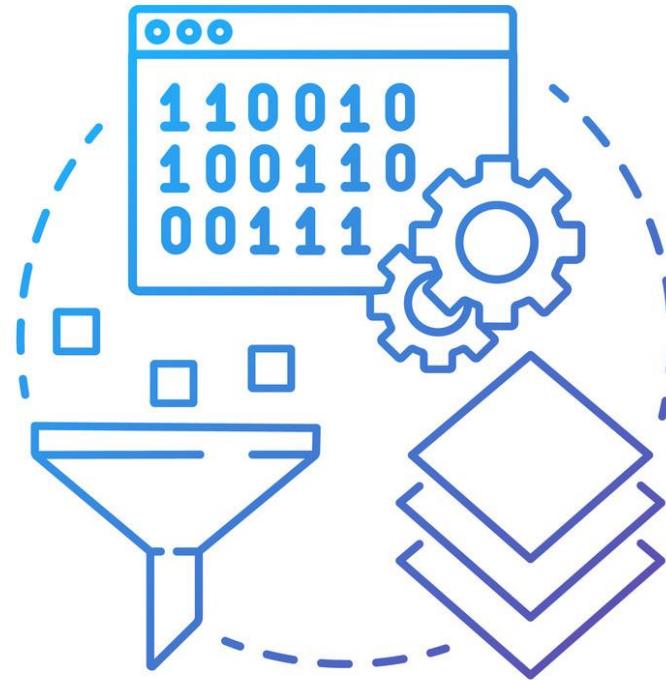


Dato curioso x2

- Python esta escrito en C



Compilador



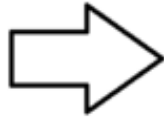
Compiler

Compiler

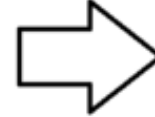
```
#include<stdio.h>

int main()
{
    printf("Hello, World!\n");
    return 0;
}
```

hello_world.c



Compiler



```
0110011000100010011000111
1100000001111111110000001
1111000110101010001100011
0011000100010011000111110
0000001111111110000001111
1000110101010001100011001
1000100010011000111110000
0001111111110000001111100
0110101010001100011001100
0100010011000111110000000
1111111110000001111100011
```

hello_world.o

Course Overview





Semanas 1-2

Introducción a la programación
Introducción a las computadoras
Lenguajes de programación



Semanas 2-9

Bases en C++
Variables, entradas y salidas
Resolución de problemas
Control de flujo, ciclos



Semanas 10-15

Vectores, Arrays y algoritmos
Modularidad
Mini Proyectos

Planificación



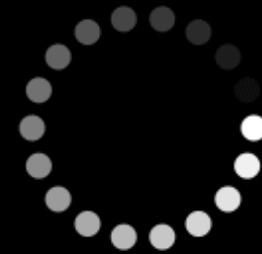
Semanas 2 - 9

Fundamentos en
programación y
resolución de problemas
mediante C++



Semanas 10 - 15

Resolución de
problemas mediante la
escritura de múltiples
files

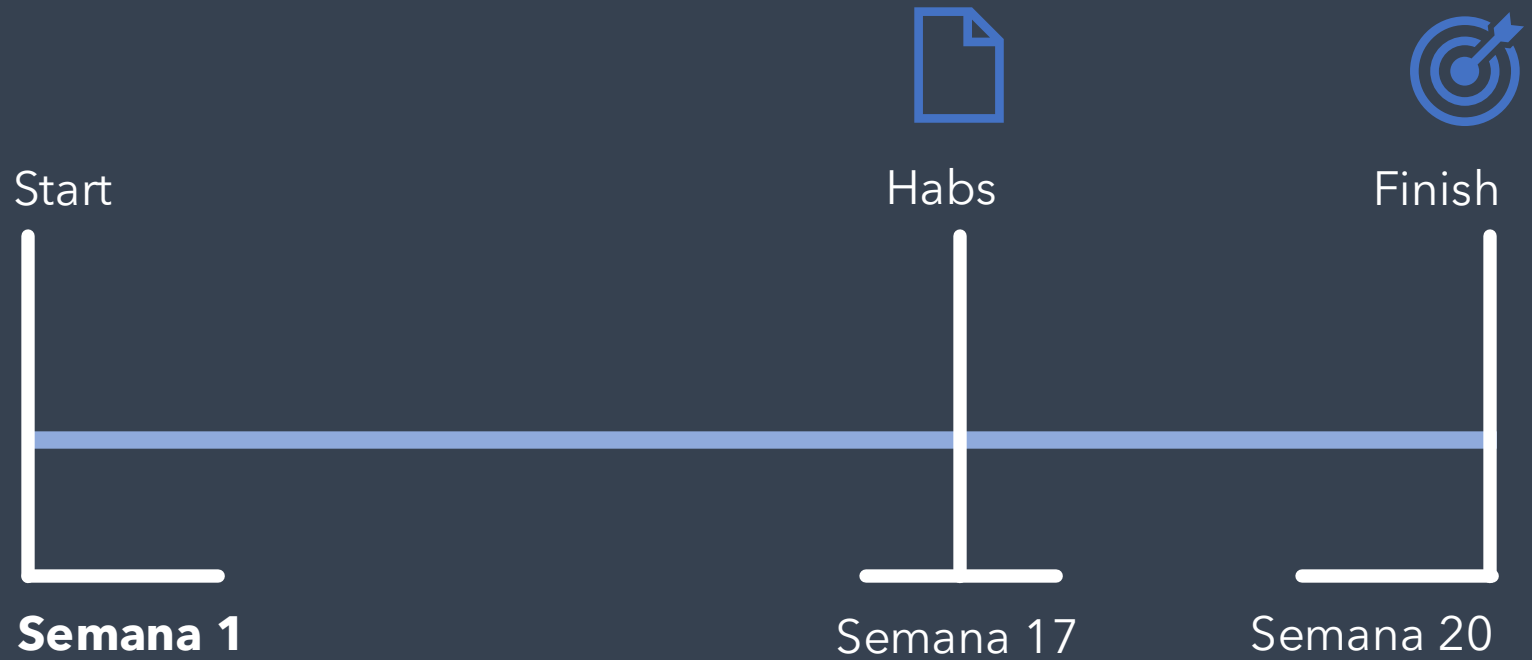


Organización

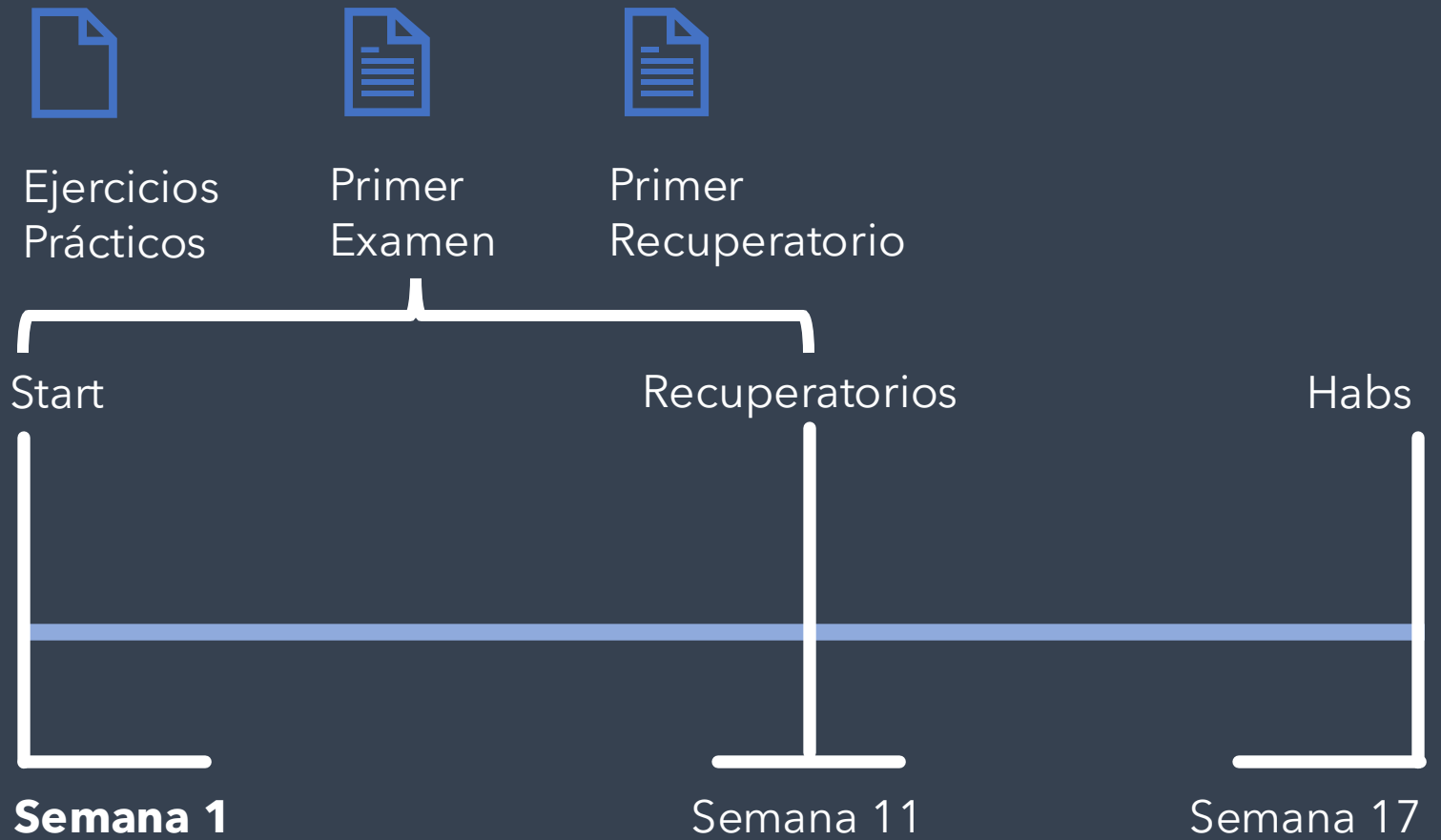


@KARONSANIMALS

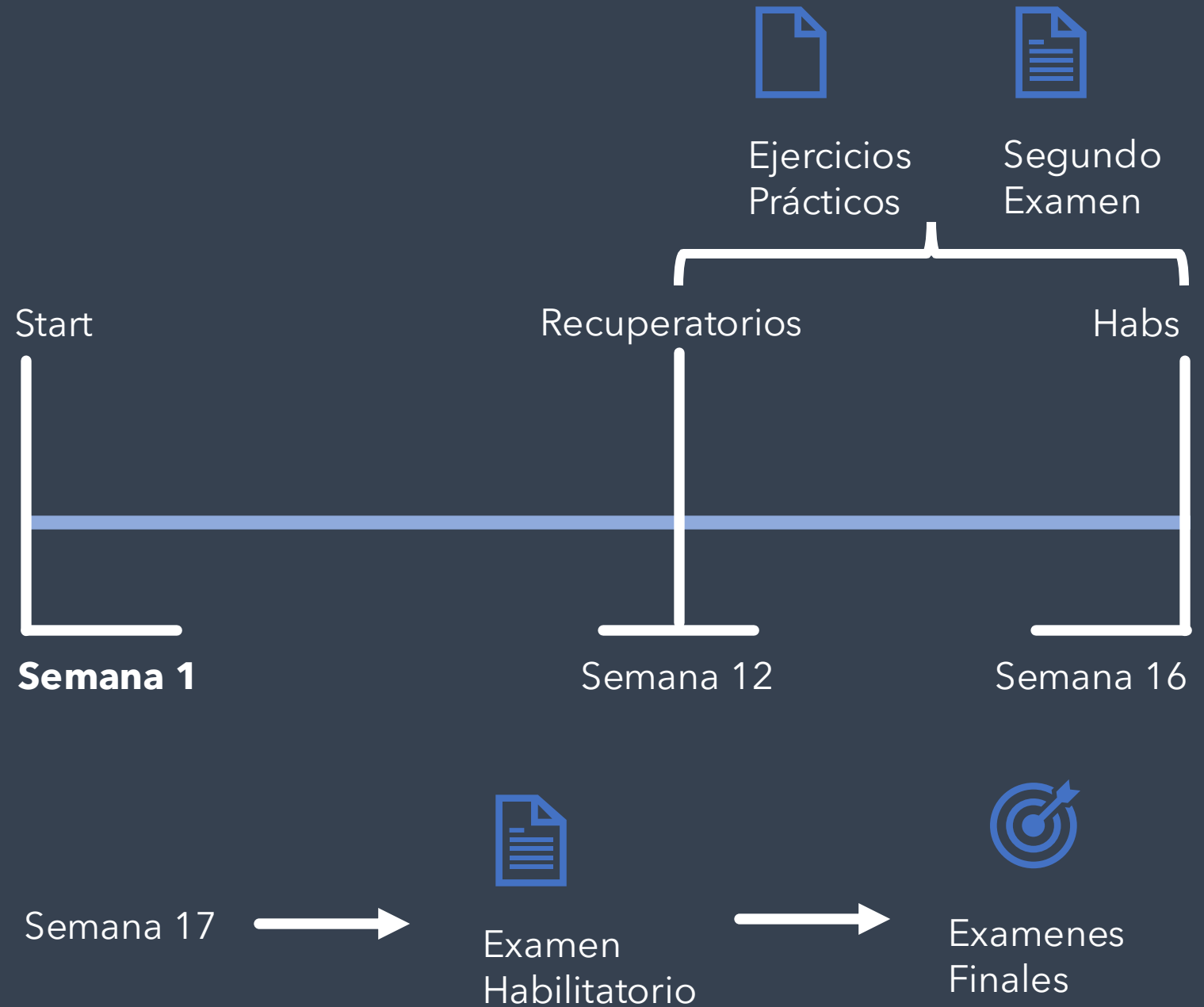
Semanas de clases



Semanas de clases



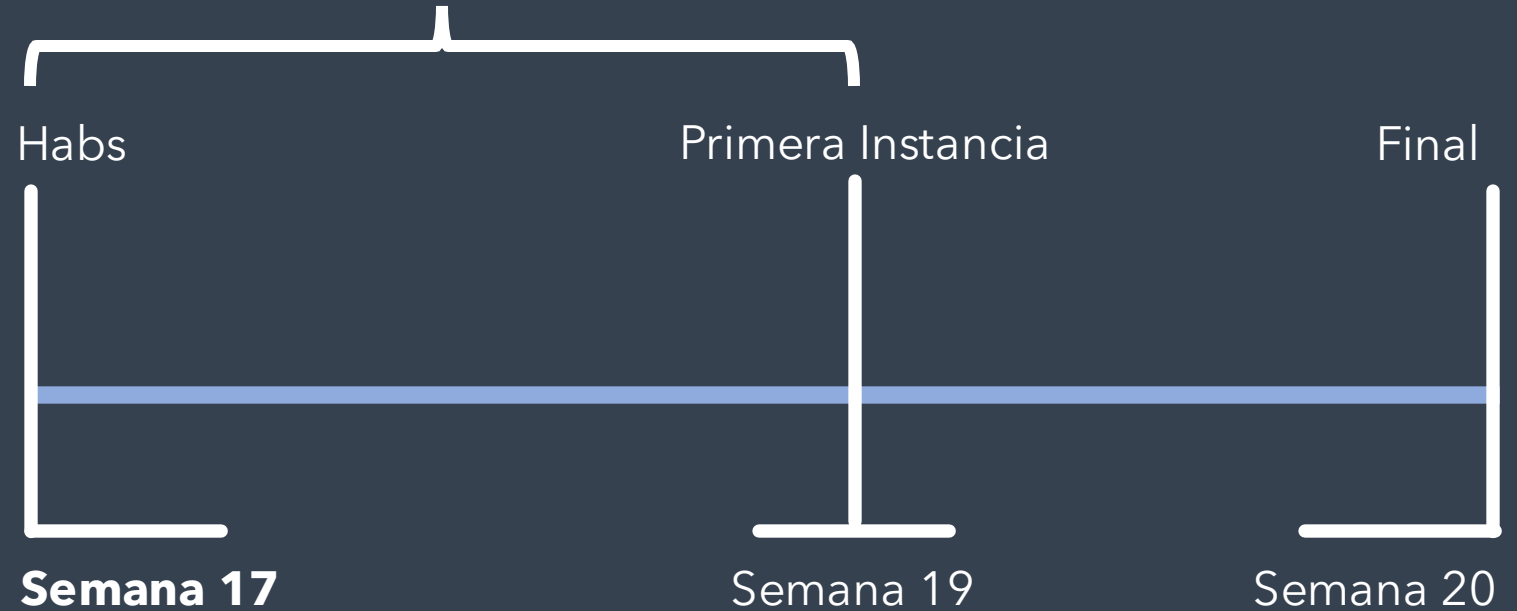
Semanas de clases



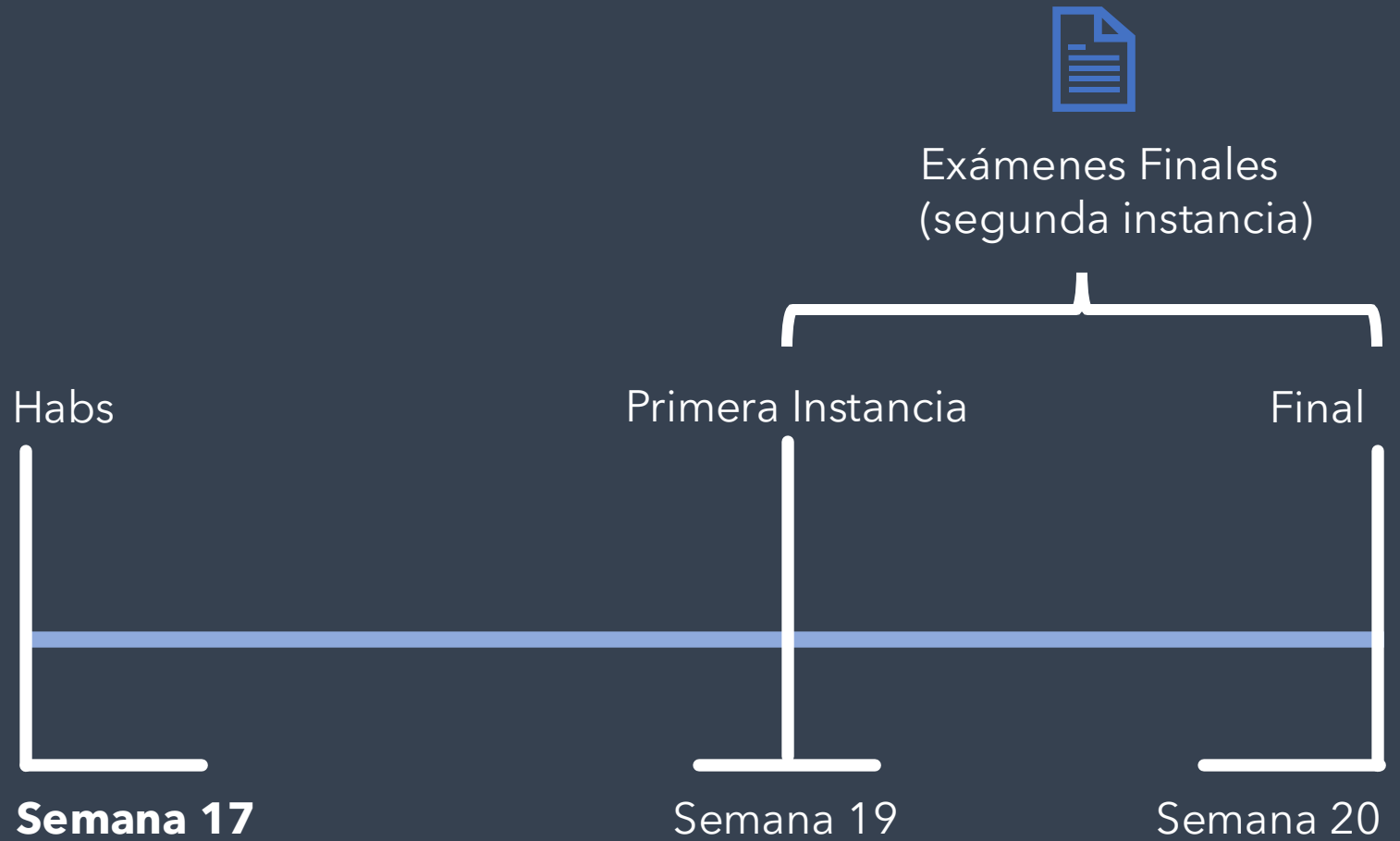
Semanas de clases



Exámenes Finales
Mini Proyectos



Semanas de clases



Estructura del Curso

Semanalmente:

- **Lecciones Teóricas**, 2 Horas académicas (Lunes 9:10 - 10:40)
- **Lecciones practicas y de revisión**, 3 Horas académicas (Viernes 9:10 - 11:30)

Practicas:

- **Ejercicios escritos** - Presentar de forma detallada y clara
- **Prácticos** - Por cada uno presentar su código y si es necesario un informe
- **Quizzes**
- **Los ejercicios no valen nota, pero se debe entregar al menos un 80% para ingresar al examen**
- **La asistencia es obligatoria, se requiere un 80% para tomar el examen**

Exámenes:

- **Examen escrito y practico**
 - Primer Examen ~ Semana 9-10
 - Segundo Examen ~ Semana 15-16

Estructura del Curso

Dos bloques

- **Bloque 1** - 100 pts
- **Bloque 2** - 100 pts

Calificaciones

- **Examen Teórico** - 40 pts de la nota del bloque
- **Examen Practico** - 60 pts de la nota del bloque.
- El examen practico tiene defensas orales.

Examen Final y Mini Proyecto:

- **Proyecto + defensas orales - 100 pts**

Importante

Clases

- **Lecciones Teóricas**
- **Lecciones prácticas y de revisión**
- Note la hora de inicio y fin de cada lección.
- Se solicita puntualidad en la clase.
- Se solicita puntualidad al momento de entregar los trabajos.
- La asistencia tiene que ser el 80 % para rendir un examen
- Las entregas de ejercicios tienen que ser del 80 % para rendir un examen

Semanas de clases



$$NotaBloque1 \geq 60$$

$$NotaBloque2 \geq 60$$

Semanas de clases

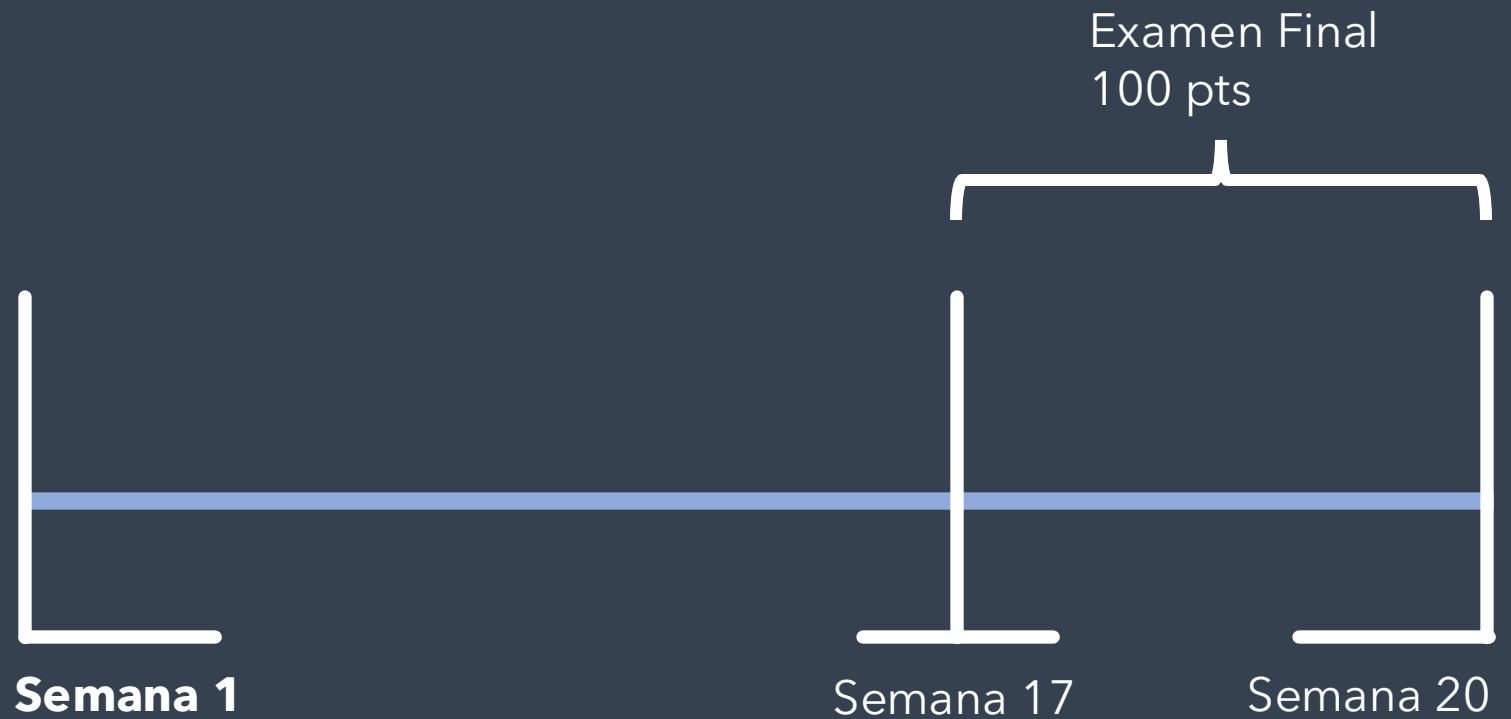


$$NotaBloque1 \geq 60$$

$$NotaBloque2 \geq 60$$

$$NotaHab = \frac{NotaBloque1 + NotaBloque2}{2}$$

Semanas de clases



$$NotaHab = \frac{NotaBloque1 + NotaBloque2}{2}$$

$$NotaFinal = \frac{NotaHab + NotaExamenFinal}{2} \geq 51$$

Practicas y reportes



Objetivo

Obtener experiencia en la resolución de problemas mediante el lenguaje de programación C++



Prácticos

Presentar los informes, códigos, ejercicios resueltos, etc.

Mandar su entrega en zip como se especifico previamente



Exámenes

Exámenes teóricos y prácticos

Herramientas

- C++ (Alguna de las siguientes)
 - Visual Studio
 - Codeblocks
 - DevC++
 - JetBrains
- Mas Avanzado:
 - **Visual Studio Code**
- Plataformas:
 - Moodle (LMS - UCB)
 - WhatsApp Grupo



Éxitos con el curso!

