

Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Clase 01: Motivación

Francisco Gazitúa Requena (fco_gazitua_requena@uc.cl)

IIC2113 Diseño Detallado de Software

05 de Agosto, 2025

¡Bienvenidos a IIC2113!

Un poco sobre mi

- Licenciado en ciencias de la ingeniería
  - Major en Ingeniería de Software
  - Minor en Data Science & Analytics

Un poco sobre mi

- Licenciado en ciencias de la ingeniería
  - Major en Ingeniería de Software
  - Minor en Data Science & Analytics
- Egresado de Ingeniería Civil de Industrias con diploma en Ingeniería de Computación

Un poco sobre mi

- Licenciado en ciencias de la ingeniería
  - Major en Ingeniería de Software
  - Minor en Data Science & Analytics
- Egresado de Ingeniería Civil de Industrias con diploma en Ingeniería de Computación
- Estudiante de Magíster en Ciencias de la Ingeniería

Un poco sobre mi

- Licenciado en ciencias de la ingeniería
  - Major en Ingeniería de Software
  - Minor en Data Science & Analytics
- Egresado de Ingeniería Civil de Industrias con diploma en Ingeniería de Computación
- Estudiante de Magíster en Ciencias de la Ingeniería
- Ayudante IIC2113 del 2023-1 al 2024-2
- Profesor IIC2113 el 2025-1

Un poco sobre mi

- Licenciado en ciencias de la ingeniería
  - Major en Ingeniería de Software
  - Minor en Data Science & Analytics
- Egresado de Ingeniería Civil de Industrias con diploma en Ingeniería de Computación
- Estudiante de Magíster en Ciencias de la Ingeniería
- Ayudante IIC2113 del 2023-1 al 2024-2
- Profesor IIC2113 el 2025-1
- Ayudante IIC2343 2024-1 e IIC3745 2025-1

Un poco sobre mi

- Licenciado en ciencias de la ingeniería
  - Major en Ingeniería de Software
  - Minor en Data Science & Analytics
- Egresado de Ingeniería Civil de Industrias con diploma en Ingeniería de Computación
- Estudiante de Magíster en Ciencias de la Ingeniería
- Ayudante IIC2113 del 2023-1 al 2024-2
- Profesor IIC2113 el 2025-1
- Ayudante IIC2343 2024-1 e IIC3745 2025-1
- Ingeniero de Software SDI

Motivación

Motivación

¿Cuál es la diferencia entre programación e ingeniería de software?

Motivación

¿Cuál es la diferencia entre programación e ingeniería de software?



Tiempo

Motivación

¿Cuál es la diferencia entre programación e ingeniería de software?



Tiempo



Escala

Motivación

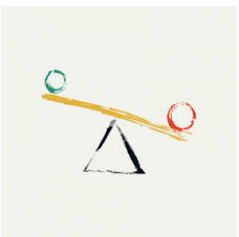
¿Cuál es la diferencia entre programación e ingeniería de software?



Tiempo



Escala



Trade-offs

Motivación

¿Cuál crees que sea la vida útil de tu código?



Motivación

¿Cuál crees que sea la vida útil de tu código?



Minutos Horas Días Semanas Meses Años Décadas

Motivación

¿Cuál crees que sea la vida útil de tu código?



Minutos Horas Días Semanas Meses Años Décadas

Problema de Programación

Motivación

¿Cuál crees que sea la vida útil de tu código?



Minutos Horas Días Semanas Meses Años Décadas

Problema de Programación

Problema de Ingeniería de Software

Motivación

¿Cuál crees que sea la vida útil de tu código?



¿Han trabajado en algún código cuya vida útil sea de 5 años o más?

Motivación

¿Cuál crees que sea la vida útil de tu código?



¿Han trabajado en algún código cuya vida útil sea de 5 años o más?

Las reglas cambian cuando tienes que crear código que vivirá por décadas

Motivación



Robert C. Martin
Clean Architecture

Motivación



Robert C. Martin
Clean Architecture

It doesn't take a huge amount of knowledge and skill to get a program working. [...] Young men and women in college start billion-dollar businesses based on scrabbling together a few lines of PHP or Ruby. [...] The code they produce may not be pretty; but it works. It works because getting something to work–once–just isn't that hard.

Motivación



Robert C. Martin
Clean Architecture

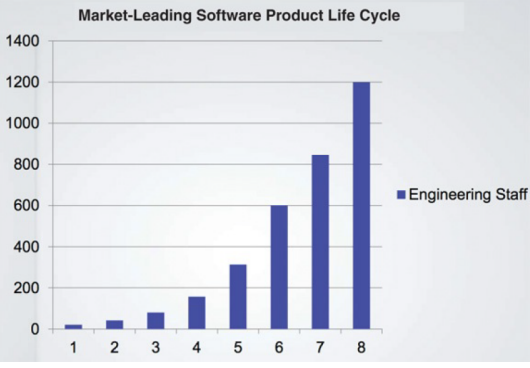
Getting it right is another matter entirely. Getting software right is hard. It takes knowledge and skills that most young programmers haven't yet aquired. [...] It requires a level of discipline and dedication that most programmers never dreamed they'd need. Mostly, it takes a passion for the craft and the desire to be a professional.

Veamos un caso[1]

[1] Robert C. Martin. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. 1st. USA: Prentice Hall Press, 2017. ISBN: 0134494164.

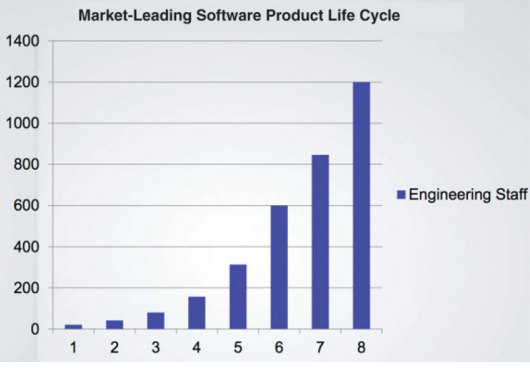
Caso

Cantidad de ingenieros v/s Release



Caso

Cantidad de ingenieros v/s Release



Tiene buena pinta

Caso

Tamaño del producto (KLOC) v/s Release



Caso

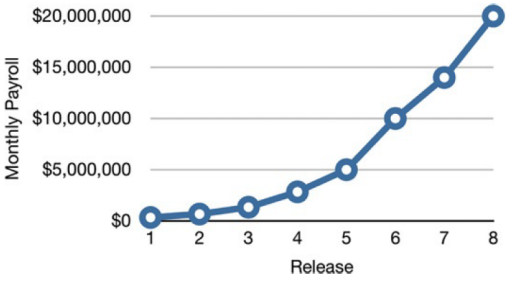
Tamaño del producto (KLOC) v/s Release



Algo anda mal

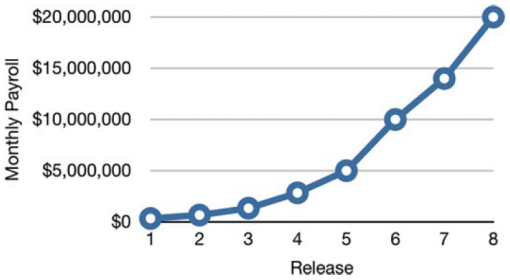
Caso

Costo mensual de sueldos v/s Release



Caso

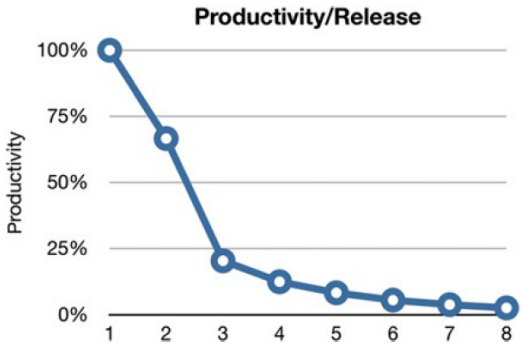
Costo mensual de sueldos v/s Release



Algo anda MUY mal!

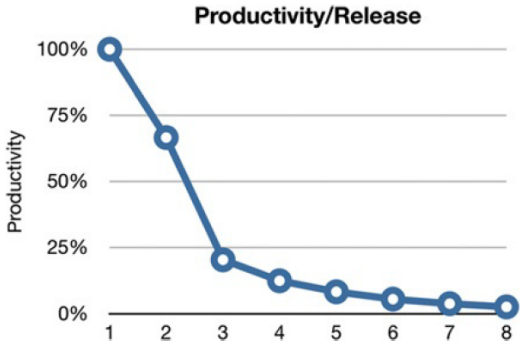
Caso

¿Que es lo que está pasando?



Caso

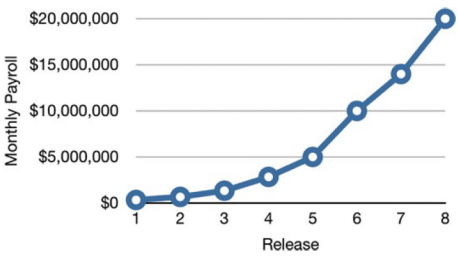
¿Que es lo que está pasando?



Esto a pesar que todos trabajan al 100%

Caso

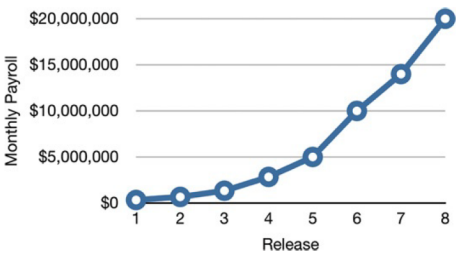
En perspectiva





Caso

En perspectiva

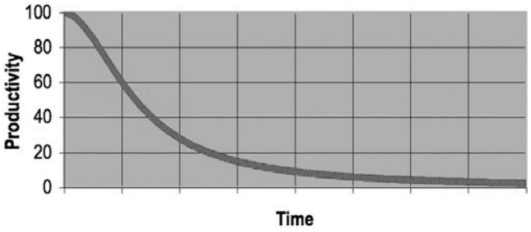




Los primeros miles de dólares compraron muchas funcionalidades, pero los últimos millones casi nada!!!

¿Que puede causar esto?

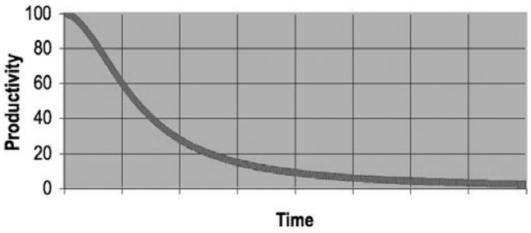
"*The Total Cost of Owning a Mess*"



Clean Code, Robert C. Martin

¿Que puede causar esto?

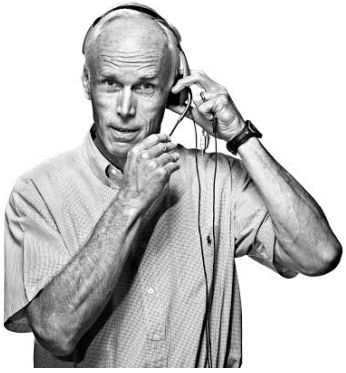
"*The Total Cost of Owning a Mess*"



Clean Code, Robert C. Martin

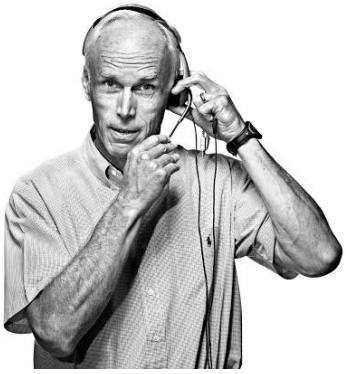
Every change [the team] make to the code breaks two or three other parts of the code. No change is trivial. Every addition or modification to the system requires that the tangles, twists, and knots be "understood" so that more tangles, twists, and knots can be added. Over time the mess becomes so big and so deep and so tall, they can not clean it up. There is no way at all.

Motivación



John Ousterhout
Stanford University

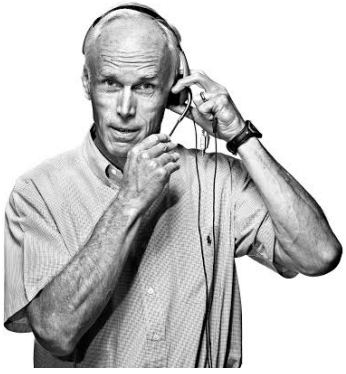
Motivación



John Ousterhout
Stanford University

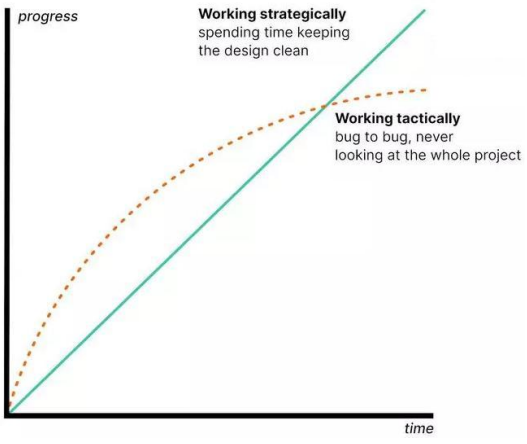
WORKING CODE ISN'T ENOUGH

Motivación



John Ousterhout
Stanford University

WORKING CODE ISN'T ENOUGH



Motivación

Objetivo: crear código que sea sustentable en el tiempo.

Motivación

Objetivo: crear código que sea sustentable en el tiempo.

Código sustentable

Un proyecto de software es sustentable si, durante toda su vida útil, somos capaces de reaccionar y adaptarnos a cualquier cambio valioso que pueda surgir, ya sea por razones técnicas o de negocio.

Motivación

Objetivo: crear código que sea sustentable en el tiempo.

Código sustentable

Un proyecto de software es sustentable si, durante toda su vida útil, somos capaces de reaccionar y adaptarnos a cualquier cambio valioso que pueda surgir, ya sea por razones técnicas o de negocio.

... este curso es un paso hacia crear código sustentable.

Motivación



Motivación





Motivación







Motivación







En este curso vamos a programar harto

Evaluaciones

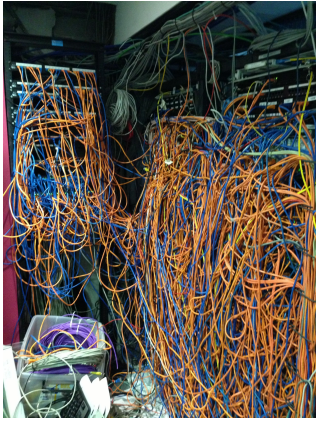
Evaluaciones

El código programado por un ingeniero de software debe ser **excelente**.

Evaluaciones

El código programado por un ingeniero de software debe ser **excelente**.





Evaluaciones

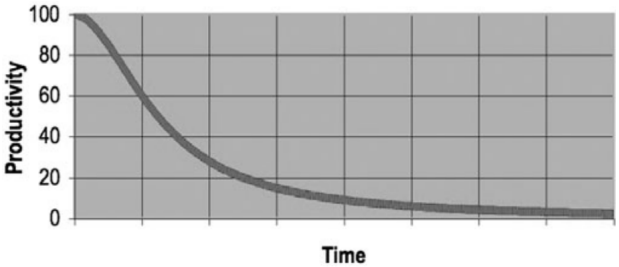
El código programado por un ingeniero de software UC debe ser **excelente**.

¿Cómo logramos esto?

Evaluaciones

El código programado por un ingeniero de software UC debe ser **excelente**.

¿Cómo logramos esto?



Necesito que sientan el peso de esta curva.

Evaluaciones

[60%] Proyecto (P):

- Tendrá 3 entregas acumulativas.
- El proyecto es individual.

Evaluaciones

[60%] Proyecto (P):

- Tendrá 3 entregas acumulativas.
- El proyecto es individual.

En cada entrega se evaluarán dos elementos principales:

1. Que el código funcione.
2. La calidad del código.

Evaluaciones

[60%] Proyecto (P):

- Tendrá 3 entregas acumulativas.
- El proyecto es individual.

En cada entrega se evaluarán dos elementos principales:

1. Que el código funcione.
2. La calidad del código.

En cada entrega les pediremos que agreguen nuevas funcionalidades sobre la entrega anterior. Si tu diseño es bueno, hacer cambios será fácil. Si tu diseño es malo (o no alcanzaste a completar la entrega anterior) hacer los cambios será difícil

Evaluaciones

Ponderación: $\mathbf{P} = 0.2 \cdot \mathbf{P}_1 + 0.3 \cdot \mathbf{P}_2 + 0.5 \cdot \mathbf{P}_3$

Evaluaciones

Ponderación: $\mathbf{P} = 0.2 \cdot \mathbf{P}_1 + 0.3 \cdot \mathbf{P}_2 + 0.5 \cdot \mathbf{P}_3$

Entrega opcional

- Habrá una cuarta entrega opcional para quienes la necesiten.

Evaluaciones

Ponderación: $\mathbf{P} = 0.2 \cdot \mathbf{P}_1 + 0.3 \cdot \mathbf{P}_2 + 0.5 \cdot \mathbf{P}_3$

Entrega opcional

- Habrá una cuarta entrega opcional para quienes la necesiten.
- En caso de realizarla, las ponderaciones serán las siguientes:

$$\mathbf{P} = 0.1 \cdot \mathbf{P}_1 + 0.2 \cdot \mathbf{P}_2 + 0.3 \cdot \mathbf{P}_3 + 0.4 \cdot \mathbf{P}_4$$

Evaluaciones

Ponderación: $\mathbf{P} = 0.2 \cdot \mathbf{P}_1 + 0.3 \cdot \mathbf{P}_2 + 0.5 \cdot \mathbf{P}_3$

Entrega opcional

- Habrá una cuarta entrega opcional para quienes la necesiten.
- En caso de realizarla, las ponderaciones serán las siguientes:

$$\mathbf{P} = 0.1 \cdot \mathbf{P}_1 + 0.2 \cdot \mathbf{P}_2 + 0.3 \cdot \mathbf{P}_3 + 0.4 \cdot \mathbf{P}_4$$

Consideraciones importantes

- En caso de no entregar por alguna razón de fuerza mayor y esta sea justificada por la **DIPRE**, la nota de esa entrega será reemplazada por la nota de la entrega final.

Evaluaciones

Ponderación: $\mathbf{P} = 0.2 \cdot \mathbf{P}_1 + 0.3 \cdot \mathbf{P}_2 + 0.5 \cdot \mathbf{P}_3$

Entrega opcional

- Habrá una cuarta entrega opcional para quienes la necesiten.
- En caso de realizarla, las ponderaciones serán las siguientes:

$$\mathbf{P} = 0.1 \cdot \mathbf{P}_1 + 0.2 \cdot \mathbf{P}_2 + 0.3 \cdot \mathbf{P}_3 + 0.4 \cdot \mathbf{P}_4$$

Consideraciones importantes

- En caso de no entregar por alguna razón de fuerza mayor y esta sea justificada por la **DIPRE**, la nota de esa entrega será reemplazada por la nota de la entrega final.
- No se aceptarán trabajos atrasados.

Evaluaciones

Ponderación: $\mathbf{P} = 0.2 \cdot \mathbf{P}_1 + 0.3 \cdot \mathbf{P}_2 + 0.5 \cdot \mathbf{P}_3$

Entrega opcional

- Habrá una cuarta entrega opcional para quienes la necesiten.
- En caso de realizarla, las ponderaciones serán las siguientes:

$$\mathbf{P} = 0.1 \cdot \mathbf{P}_1 + 0.2 \cdot \mathbf{P}_2 + 0.3 \cdot \mathbf{P}_3 + 0.4 \cdot \mathbf{P}_4$$

Consideraciones importantes

- En caso de no entregar por alguna razón de fuerza mayor y esta sea justificada por la **DIPRE**, la nota de esa entrega será reemplazada por la nota de la entrega final.
- No se aceptarán trabajos atrasados.
- Habrán salas de ayuda durante el semestre

Evaluaciones

Entregas del proyecto:

- **[03/09 a las 23:59]**: Entrega 1 proyecto.
- **[08/10 a las 23:59]**: Entrega 2 proyecto.
- **[05/11 a las 23:59]**: Entrega 3 proyecto.
- **[03/12 a las 23:59]**: Entrega 4 proyecto.

Política de entregas atrasadas:

Evaluaciones

Entregas del proyecto:

- **[03/09 a las 23:59]**: Entrega 1 proyecto.
- **[08/10 a las 23:59]**: Entrega 2 proyecto.
- **[05/11 a las 23:59]**: Entrega 3 proyecto.
- **[03/12 a las 23:59]**: Entrega 4 proyecto.

Política de entregas atrasadas: No hay

Evaluaciones

Entregas del proyecto:

- **[03/09 a las 23:59]**: Entrega 1 proyecto.
- **[08/10 a las 23:59]**: Entrega 2 proyecto.
- **[05/11 a las 23:59]**: Entrega 3 proyecto.
- **[03/12 a las 23:59]**: Entrega 4 proyecto.

Política de entregas atrasadas: No hay

- Canvas se cierra y no se aceptan más entregas.

Evaluaciones

Entregas del proyecto:

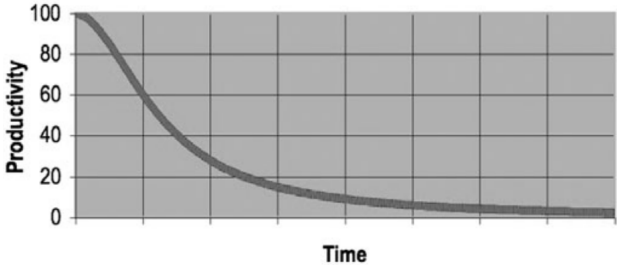
- **[03/09 a las 23:59]**: Entrega 1 proyecto.
- **[08/10 a las 23:59]**: Entrega 2 proyecto.
- **[05/11 a las 23:59]**: Entrega 3 proyecto.
- **[03/12 a las 23:59]**: Entrega 4 proyecto.

Política de entregas atrasadas: No hay

- Canvas se cierra y no se aceptan más entregas.
- En caso de enfermedad (o similar), si la DIPRE te justifica una entrega entonces esa entrega es reemplazada por la nota en la $\mathbf{P}_4$.

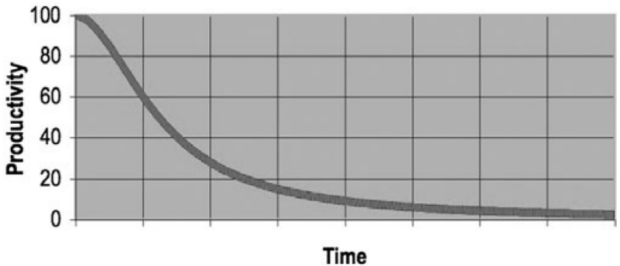
Evaluaciones

Con el proyecto sentirán el peso de esta curva (debido a tu propio código)



Evaluaciones

Con el proyecto sentirán el peso de esta curva (debido a tu propio código)



Además los ayudantes revisarán que tu código sea **excelente**.

Evaluaciones

[10%] Nota de cátedra (C):

Evaluaciones

[10%] Nota de cátedra (C):

- Son ejercicios (o cuestionarios) que debes resolver entre clases.

Evaluaciones

[10%] Nota de cátedra (C):

- Son ejercicios (o cuestionarios) que debes resolver entre clases.
- Algunos serán revisados mediante evaluación por pares en Canvas.

Evaluaciones

[10%] Nota de cátedra (C):

- Son ejercicios (o cuestionarios) que debes resolver entre clases.
- Algunos serán revisados mediante evaluación por pares en Canvas.
  - Cuando subas tu solución te serán asignadas n tareas de compañeros.
  - Tendrás una semana para revisar las tareas siguiendo una rúbrica simple.

Evaluaciones

[10%] Nota de cátedra (C):

- Son ejercicios (o cuestionarios) que debes resolver entre clases.
- Algunos serán revisados mediante evaluación por pares en Canvas.
  - Cuando subas tu solución te serán asignadas n tareas de compañeros.
  - Tendrás una semana para revisar las tareas siguiendo una rúbrica simple.
  - Si no revisas las tareas de tus compañeros tendrás cero puntos en esa actividad.

Evaluaciones

[10%] Nota de cátedra (C):

- Son ejercicios (o cuestionarios) que debes resolver entre clases.
- Algunos serán revisados mediante evaluación por pares en Canvas.
  - Cuando subas tu solución te serán asignadas n tareas de compañeros.
  - Tendrás una semana para revisar las tareas siguiendo una rúbrica simple.
  - Si no revisas las tareas de tus compañeros tendrás cero puntos en esa actividad.
  - **Se podrá penalizar la evaluación de pares incorrecta. Por ejemplo, ponerle todo el puntaje a una tarea que no compila**.

Evaluaciones

[10%] Nota de cátedra (C):

- Son ejercicios (o cuestionarios) que debes resolver entre clases.
- Algunos serán revisados mediante evaluación por pares en Canvas.
  - Cuando subas tu solución te serán asignadas n tareas de compañeros.
  - Tendrás una semana para revisar las tareas siguiendo una rúbrica simple.
  - Si no revisas las tareas de tus compañeros tendrás cero puntos en esa actividad.
  - **Se podrá penalizar la evaluación de pares incorrecta. Por ejemplo, ponerle todo el puntaje a una tarea que no compila**.
- Al final del semestre sumaremos todos los puntos que obtuviste en las actividades y los escalaremos para obtener la nota de cátedra.

Evaluaciones

[10%] Nota de cátedra (C):

- Son ejercicios (o cuestionarios) que debes resolver entre clases.
- Algunos serán revisados mediante evaluación por pares en Canvas.
  - Cuando subas tu solución te serán asignadas n tareas de compañeros.
  - Tendrás una semana para revisar las tareas siguiendo una rúbrica simple.
  - Si no revisas las tareas de tus compañeros tendrás cero puntos en esa actividad.
  - **Se podrá penalizar la evaluación de pares incorrecta. Por ejemplo, ponerle todo el puntaje a una tarea que no compila**.
- Al final del semestre sumaremos todos los puntos que obtuviste en las actividades y los escalaremos para obtener la nota de cátedra.
- Adicionalmente podrán haber actividades que den puntos extras.

Evaluaciones

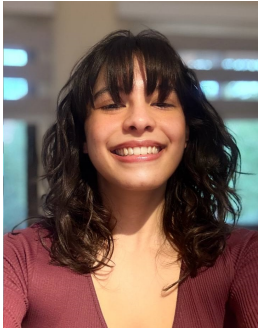
[10%] Nota de cátedra (C):

- No se aceptarán tareas o revisiones de pares atrasadas en ninguna circunstancia

Evaluaciones

[10%] Nota de cátedra (C):

- No se aceptarán tareas o revisiones de pares atrasadas en ninguna circunstancia
- Ante cualquier duda o problema con su nota de cátedra, contactar a la **Coordinadora de Cátedra**



Javiera Pinto
jpints@uc.cl

Evaluaciones

El proyecto evalúa que tu código sea excelente. Falta evaluar que puedas mejorar el código escrito por alguien más.

[15%] Interrogación (I):

- **Fecha**: 22/10 a las 17:30
- Será en computador.
- Deberán mejorar un código dado por nosotros (limpieza de código).

[15%] Examen (E):

- **Fecha**: 10/12 a las 17:30
- Será en computador.
- Deberán mejorar un código dado por nosotros (patrones de diseño).

Evaluaciones

Criterios de aprobación:

Evaluaciones

Criterios de aprobación:

- **NF** ≥ 3.95

Evaluaciones

Criterios de aprobación:

- **NF** ≥ 3.95
- **E** ≥ 2.95

Evaluaciones

Criterios de aprobación:

- **NF** ≥ 3.95
- **E** ≥ 2.95

Puntos a considerar:

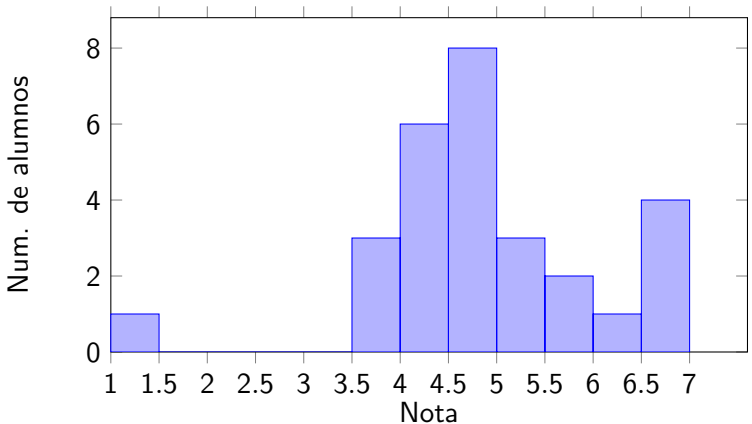
- El proyecto y la interrogación no son reprobatorios
- Si deciden entregar la E4, su nota final se calculará utilizándola aunque esto les baje el promedio
- Si su nota del examen es menor a 2.95 reprobarán el ramo independiente de si su nota final es mayor o igual a 3.95
- Sin embargo, sacarse un 3 en el examen ha sido en general fácil

Evaluaciones

Veamos las notas del último examen

Evaluaciones

Veamos las notas del último examen



En ese semestre el 96% de los estudiantes cumplieron el criterio del examen y es el examen más difícil que hemos tenido

Evaluaciones

Resumen:

- **[60%]** Proyecto de curso.
- **[10%]** Nota de cátedra.
- **[15%]** Interrogación.
- **[15%]** Examen.

Criterio aprobación: $\mathbf{NF} \geq 3.95 \wedge \mathbf{E} \geq 2.95$.

Evaluaciones

Resumen:

- **[60%]** Proyecto de curso.
- **[10%]** Nota de cátedra.
- **[15%]** Interrogación.
- **[15%]** Examen.

Criterio aprobación: $\mathbf{NF} \geq 3.95 \wedge \mathbf{E} \geq 2.95$.

Con esto, buscamos garantizar dos cosas:

- Puedes programar código que sea excelente.
- Puedes mejorar y construir sobre código existente.

¿Dudas?