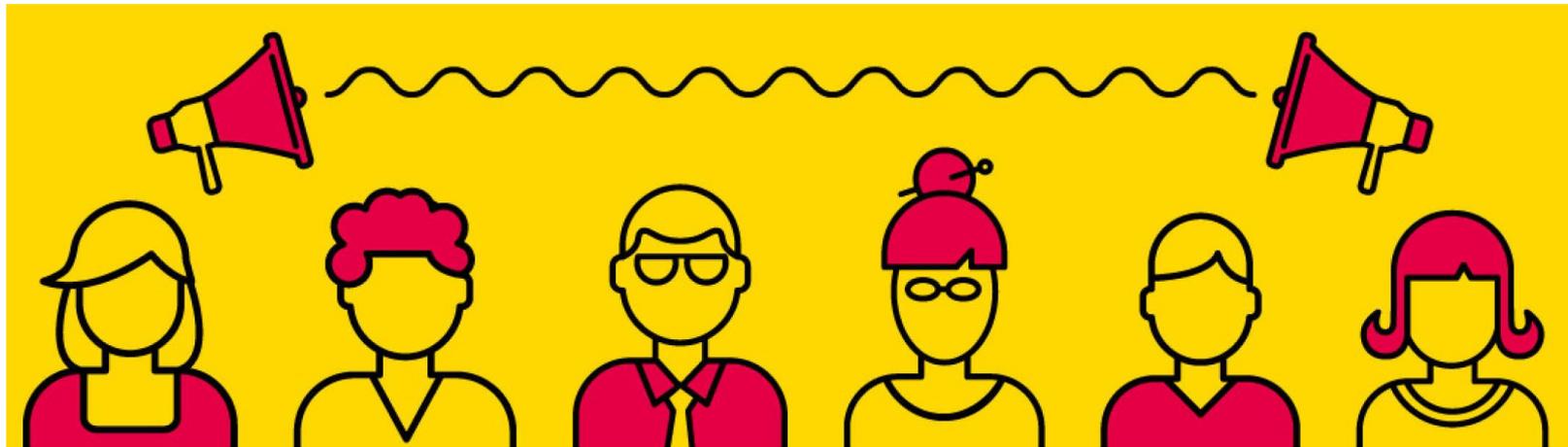


# Ingeniería de Software Basada en Evidencia

## Clase 1

# Encuesta Inicial

---



# Información General

---

- Clases martes 18 a 19.30hs y un monitoreo de 20 min por semana.
- Modalidad taller: lectura previa a la clase y trabajo en clase.
  - Control de lectura en cada clase.
- Eva del curso: <https://eva.fing.edu.uy/course/view.php?id=1086>
- Libro del curso:
  - Evidence-based software engineering and systematic reviews – Kitchenham, Budgen & Brereton, 2015.
  - Además hay un reporte técnico introductorio
- Evaluación: trabajo grupal 50%, eval. y pruebas individuales 50%

# Calendario del curso

---

<u>Sem/Clase</u>	<u>Tema</u>	<u>Evaluaciones?</u>
1 08/08	Introducción a la investigación en SE	Preguntas EVA
2 15/08	Práctica basada en evidencia	Preguntas EVA
3 22/08	Planificación de una SR	Preguntas EVA
4 29/08	Búsqueda de estudios	Preguntas EVA Avance Trabajo
5 05/09	Selección de estudios	Preguntas EVA Avance Trabajo
6 12/09	Evaluación de calidad	Preguntas EVA Avance Trabajo
7 03/10	Extracción de datos	Preguntas EVA Avance Trabajo
8 10/10	Mapeos sistemáticos	Preguntas EVA Avance Trabajo
9 17/10	Síntesis cualitativa	Evaluación Ind* Avance Trabajo
10 24/10	Reporte de SRs	Preguntas EVA Avance Trabajo
11 31/10	Traducción de conocimiento	Evaluación Ind* Avance Trabajo
12 07/11	Uso de evidencia y consideraciones	Preguntas EVA Avance Trabajo
13 14/11	*Libre*	
14 21/11	Prueba final y Entrega de trabajo grupal	Evaluación Ind. Final

Evaluación Ind\* → Lectura de un artículo científico y luego responder 3 preguntas en EVA

---

# Información General

---

- Un trabajo en grupo a realizar durante el curso.
- Unidad curricular guiada por Objetivos de aprendizaje (LOs).

## **Aspectos básicos de publicaciones científicas**

---

2. Interpretar las distintas secciones de un artículo científico
3. Acceder a artículos científicos mediante bibliotecas digitales y motores de búsqueda
4. Diferenciar entre literatura científica con referato, literatura gris, publicaciones de divulgación científica y artículos de opinión

## **Investigación en ingeniería de software**

---

5. Describir el proceso de investigación científica
6. Describir la necesidad de conocimiento científico en ingeniería de software
7. Comparar diferentes métodos de investigación utilizados en ingeniería de software
8. Participar en la identificación de métodos de investigación utilizados en artículos científicos

# Comentarios de estudiantes

---

- *Selección de tema del trabajo → muy importante, o que lo hagan los docentes o que traten de elegir temas que conozcan un poco, vienen dificultades luego.*
- *Lecturas previas → arrancar por leer el reporte técnico, el 40% de los estudiantes sugiere incorporar controles de lectura.*
- *Tratar de participar más en el foro, no dejar todas las dudas para las clases.*
- *Agregar puntaje a los controles de lectura → ayudan a estar al día y motiva a que se realice la lectura previa a la clase*

# ¿Qué usamos como apoyo en la toma de decisiones?

- experiencia personal
- charlar con colegas
- leer blogs o foros
- leer libros
- asistir a eventos o talleres
- ver qué hacen los demás
- leer revistas especializadas



# Usabilidad de la evidencia



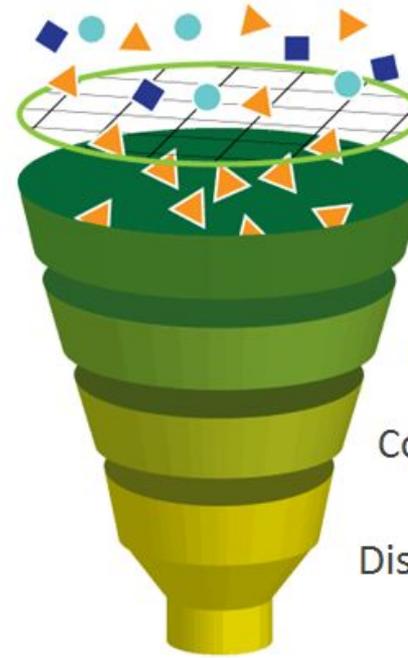
# Pasos de EBSE

- 1) Convertir el problema o la necesidad de información en una pregunta que se pueda responder.
- 2) Buscar en la literatura la mejor evidencia disponible para responder esa pregunta. (y crear evidencia local, si es necesario)
- 3) Evaluar críticamente la evidencia en su validez, el impacto y aplicabilidad.
- 4) Integrar la evidencia evaluada con la experiencia práctica y los valores del cliente para tomar decisiones sobre la práctica.
- 5) Evaluar la eficacia y eficiencia de los pasos previos y buscar maneras de mejorarlos.

Identificar problema y preguntas

Elaborar un plan (protocolo)

Buscar estudios



Filtrar y seleccionar estudios

Extraer datos de los estudios

Evaluar la calidad de los estudios

Combinar los datos (síntesis)

Discutir y concluir resultados generales

Revisión sistemática

Difusión

## Proceso de una SLR

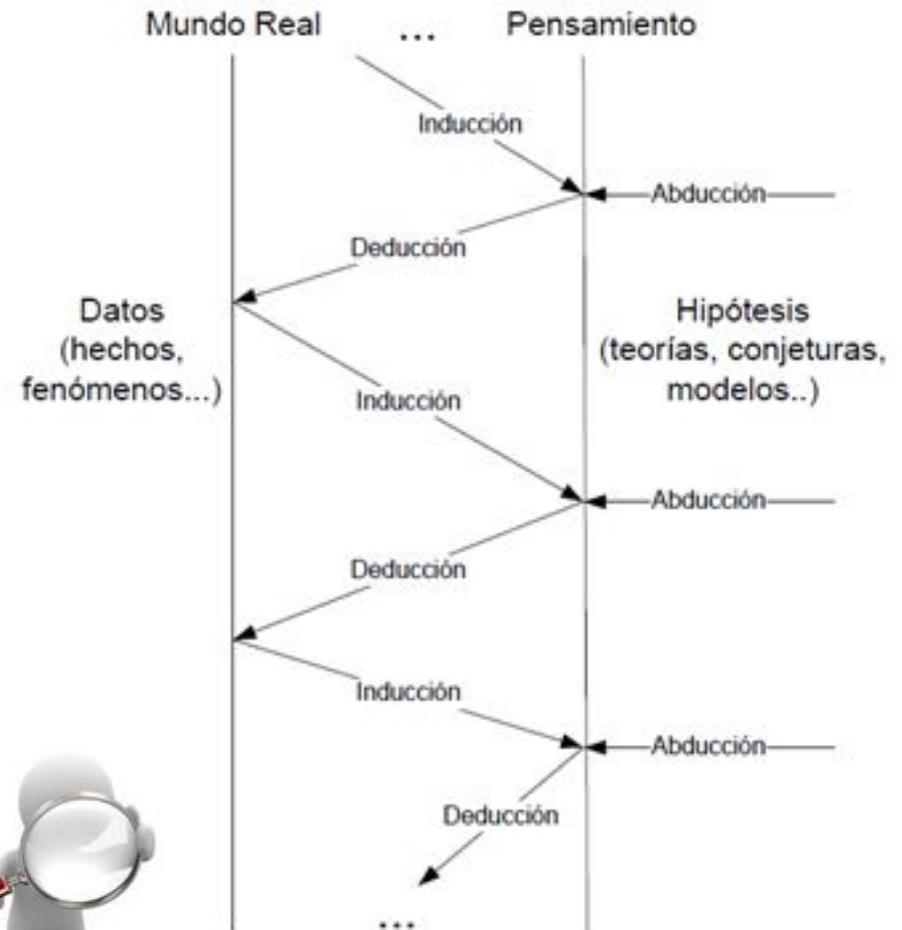
# Investigación en Ingeniería de Software

¿Qué es?

# Proceso de investigación científica

• El proceso de investigación más general está compuesto por ciclos de inducción y deducción

- La **inducción** a través de la observación de un fenómeno en la vida real
- La **deducción** a través de la teoría
- La generación de nuevas hipótesis o conjeturas que puedan ser comprobadas se conoce como **abducción**



# Proceso de investigación científica

---

Un esquema deductivo:

- Regla general: "Todos los frijoles del empaque X son blancos".
- Caso: "Estos frijoles provienen del empaque X".
- Resultado: "Estos frijoles son blancos".

Un esquema inductivo:

- Caso: "Estos frijoles provienen del empaque X".
- Resultado: "Estos frijoles son blancos".
- Regla general: "Todos los frijoles del empaque X son blancos".

Un esquema abductivo:

- Resultado: "Estos frijoles son blancos".
- Regla general: "Todos los frijoles del empaque X son blancos".
- Caso: "Estos frijoles provienen del empaque X".

# Proceso de investigación científica

---

## Investigación en la vida diaria

### Variables:

- Ingredientes (crudos, cocidos, precocidos)
- Proceso de elaboración (tiempos, potencia de cocción, lapsos de reposo, tratamiento de los ingredientes, orden de procesamiento)
- La percepción de “rico” (subjetivo)

¡Cuando cocinamos!



# Investigación en Ingeniería de Software

---

- Objetivos
  - Generar conocimiento, comprender, **predecir** comportamiento
  - Beneficios (en la industria): aumentar la productividad, mejorar la calidad, aumentar la eficiencia
- ¿Qué se investiga?
  - Técnicas, procesos, métodos, modelos
- Diferencias con otras disciplinas:
  - Mezcla elementos sociales
  - Es inmadura en comparación con otras disciplinas como la química, física o biología
  - La Ingeniería de Software crea sus propios objetos y fenómenos de estudio

# Inmadurez de Ing. de Software (ejemplos de 2023)

**Tesla reparará más de un millón de coches en China por varios problemas de software**

TECNOLOGÍA / APPLE

**Apple confirma que los iPhone 14 Pro tienen un problema de software**

Firma japonesa dice que su módulo lunar chocó por fallas de software y cambio de ubicación

**El Pentágono suspende las entregas de cazas F-35 TR-3 mejorados por problemas de software**

---

# From Organizations to Individuals: Psychoactive Substance Use By Professional Programmers

**Abstract**—Psychoactive substances, which influence the brain to alter perceptions and moods, have the potential to have positive and negative effects on critical software engineering tasks. They are widely used in software, but that use is not well understood. We present the results of the first qualitative investigation of the experiences of, and challenges faced by, psychoactive substance users in professional software communities. We conduct a thematic analysis of hour-long interviews with 26 professional programmers who use psychoactive substances at work. Our results provide insight into individual motivations and impacts, including mental health and the relationships between various substances and productivity. Our findings elaborate on socialization effects, including soft skills, stigma, and remote work. The analysis also highlights implications for organizational policy, including positive and negative impacts on recruitment and retention. By exploring individual usage motivations, social and cultural ramifications, and organizational policy, we demonstrate how substance use can permeate all levels of software development.

TABLE II

NUMBER OF PARTICIPANTS IN OUR SAMPLE THAT HAVE USED EACH PSYCHOACTIVE SUBSTANCE WHILE COMPLETING SOFTWARE TASKS

Substance	# of Participants
Prescription Stimulants (e.g., Adderall, Ritalin)	21
Cannabis (e.g., Marijuana, Weed)	14
Alcohol (e.g., Beer, Wine)	13
Mood Disorder Medication (e.g., SSRIs, Wellbutrin)	11
Psychedelics (e.g., LSD, Psilocybin)	7
Tobacco (e.g., Cigarettes, Vapes)	7
Cocaine	3
Benzodiazepines (e.g., Xanax)	2
Opiates (e.g., Codeine)	2

TABLE III  
POSITIVE (↑), NEGATIVE (↓) AND NEUTRAL (−) ASSESSMENTS OF SUBSTANCES ON TASKS OR ATTRIBUTES OF THE DEVELOPMENT PROCESS.

Task	<i>n</i>	Alcohol	Cannabis	Stimulants
Brainstorming	16	n/a	↑ (86%)	− (50%)
Coding & Testing	14	n/a	− (50%)	↑ (91%)
Data Analysis	11	n/a	↓ (33%)	↑ (88%)
Debugging	21	↑ (100%)	↓ (25%)	↑ (93%)
Design	15	n/a	↓ (40%)	− (58%)
Documentation	14	n/a	↓ (25%)	↑ (70%)
Meetings	12	n/a	↓ (38%)	− (50%)
Requirements Elicit.	5	n/a	n/a	n/a
<b>Individual Attributes</b>				
Creativity	13	↑ (100%)	↑ (100%)	↓ (25%)
Enjoyment of Work	12	↑ (100%)	↑ (89%)	↑ (80%)
Focus & Productivity	25	↑ (60%)	↓ (33%)	↑ (95%)
Quality of Work	12	↓ (33%)	↓ (40%)	n/a
<b>Social Attributes</b>				
“Soft” Skills	18	n/a	↓ (40%)	↑ (80%)

*n* counts participants that mention the task in conjunction with a substance. Percentages give the fraction of positive and negative mentions that were positive. Combinations with fewer than four mentions are not analyzed (“n/a”); positive or negative outcomes with 10 or more mentions are bolded.

# Lessons from Eight Years of Operational Data from a Continuous Integration Service

An Exploratory Case Study of CircleCI

---

## ABSTRACT

Continuous Integration (CI) is a popular practice that enables the rapid pace of modern software development. Cloud-based CI services have made CI ubiquitous by relieving software teams of the hassle of maintaining a CI infrastructure. To improve these CI services, prior research has focused on analyzing historical CI data to help service consumers. However, finding areas of improvement for CI service providers could also improve the experience for service consumers. To search for these opportunities, we conduct an empirical study of 22.2 million builds spanning 7,795 open-source projects that used CIRCLECI from 2012 to 2020.

First, we quantitatively analyze the builds (i.e., invocations of the CI service) with passing or failing outcomes. We observe that the heavy and typical service consumer groups spend significantly different proportions of time on seven of the nine build actions (e.g., dependency retrieval). On the other hand, the compilation and testing actions consistently consume a large proportion of build time across consumer groups (median 33%). Second, we study builds that terminate prior to generating a pass or fail signal. Through a systematic manual analysis, we find that availability issues, configuration errors, user cancellation, and exceeding time limits are key reasons that lead to premature build termination.

Our observations suggest that (1) heavy service consumers would benefit most from build acceleration approaches that tackle long build durations (e.g., skipping build steps) or high throughput rates (e.g., optimizing CI service job queues), (2) efficiency in CI pipelines can be improved for most CI consumers by focusing on the compilation and testing stages, and (3) avoiding misconfigurations and tackling service availability issues present the largest opportunities for improving the robustness of CI services.

## • Algunos resultados:

*Approaches to make testing and compiling faster will benefit a large proportion of CI consumers. The heaviest consumers (and CI providers as a consequence) will benefit most from approaches to optimize dependency installation.*

*Availability issues, configuration errors, user cancellation, and exceeding time limits are key reasons that lead to non-signal-generating builds. Approaches to increase the availability and improve the robustness of CI configuration will likely yield the largest reductions in non-signal-generating builds.*

---

# Contextualización de la investigación

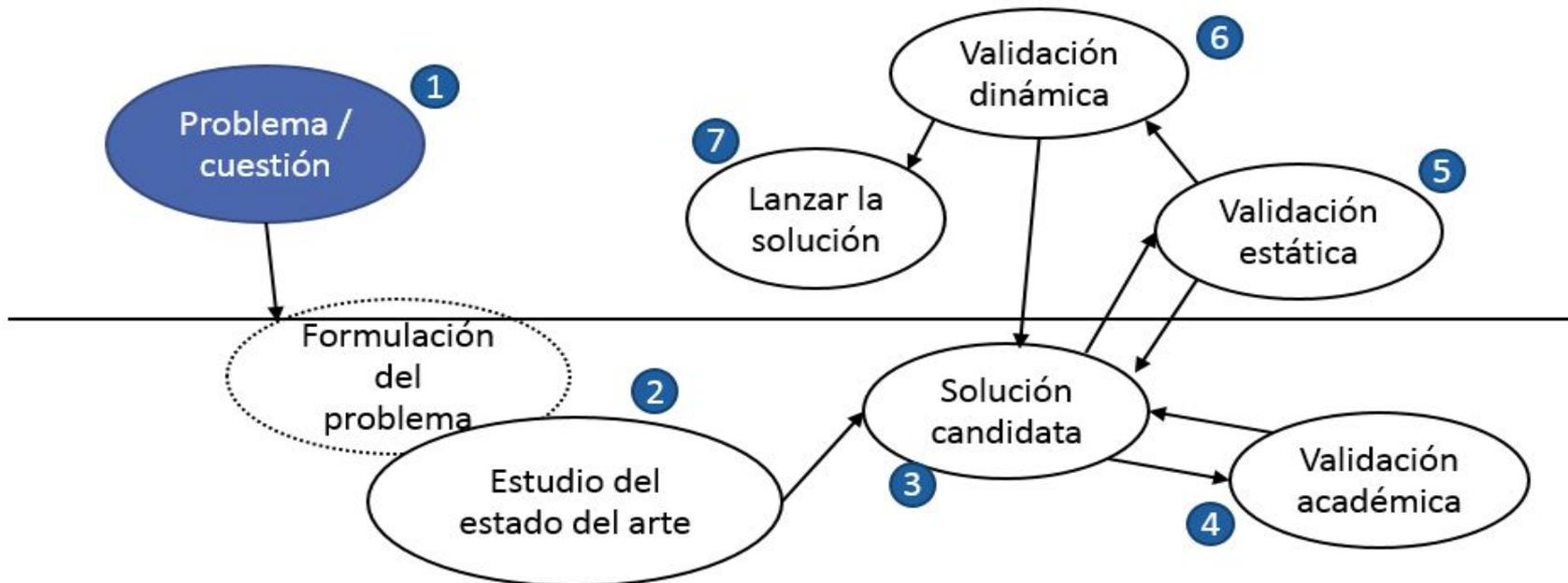
---

- Determinar el alcance de la aplicabilidad de los resultados

Contexto "ómnibus"				
¿Qué? - Fenómeno	¿Quién? - Sujeto	¿Dónde? - Ubicación	¿Cuándo? - Tiempo	¿Por qué? - Razón fundamental
Contexto Discreto				
<u>Técnico</u> Complejidad Tecnología Tarea/sistema ...	<u>Social</u> Habilidad Individual Autonomía del equipo Estructura de la organización ...	<u>Ambiental</u> Incertidumbre Comunidad Mercado ...		

# Necesidad de conocimiento científico en Ing. de Soft.

- Reducir el *gap* que existe entre la teoría y la práctica.
- Objetivos de una disciplina científica aplicada:
  - Aumentar el cuerpo de conocimiento (teórico) para entender por qué
  - Mejorar técnicas (práctico) para tener resultados útiles y aplicables.
- Posible desconexión entre el mundo teórico y el práctico
- Modelo de transferencia de tecnología en ing. sw. (Gorschek et al.: 2008)



# Metodología de Investigación

---

- Clasificación:
  - **Primarios**: métodos utilizados para realizar estudios con el objetivo de obtener evidencia empírica sobre un tema de interés
  - **Secundarios**: métodos que permiten recopilar de manera sistemática y rigurosa los estudios primarios relacionados con una pregunta de investigación específica, con el objetivo de sintetizar la evidencia disponible para responder a dicha pregunta
  
- Tipos de métodos de investigación
  - Encuestas (**primario**)
  - Experimentos (**primario**)
  - *Estudios de caso* (**primario**)
  - *Investigación-Acción (Action research)* (**primario**)
  - Revisiones sistemáticas (**secundario**)

# Algunos Métodos de Investigación primarios

---

- Encuestas: recolección de información estandarizada de una población específica o una muestra representativa, a través de un cuestionario.
- Experimentos: prueba determinadas hipótesis establecidas midiendo el efecto de manipular una variable (independiente) en otra variable (dependiente), habiendo controlado las posibles variables perturbadoras.
- Estudios de caso: son estudios observacionales, no hay aleatoriedad de variables perturbadoras, ni representatividad de la muestra. Se basa en varias fuentes de evidencia para investigar una instancia o un número pequeño de ellas de un fenómeno en su contexto real, no estando claramente definidos los límites entre el fenómeno y el contexto.
- Investigación acción: pretende cambiar el fenómeno estudiado y en el que es fundamental la inmersión del investigador en la realidad que se está investigando

# Proceso Típico de la Investigación

---



# Artículos Científicos – Reporte de resultados

---

- Los resultados pueden ser reportados a través de:
  - Artículos de conferencia
  - Artículos de revistas
  - Reportes técnicos
- Utilizados para sintetizar investigaciones y presentarlos en revistas, conferencias u otros lugares.
  - Breves (4 a 20 páginas o más), en conferencias son más cortos.
  - Involucran, en general, revisión por pares.
  - Aunque hay distintos estilos, comúnmente se tiene secciones similares:
    - Título
    - Autores (Afiliación)
    - Abstract (Resumen)
    - Palabras clave (keywords)
    - Introducción
    - Contexto o Trabajo Previo
    - Método
    - Resultados
    - Limitaciones
    - Discusión
    - Conclusiones
    - Referencias

# Estructura de artículo científico

Título

## In Search of the Emotional Design Effect in Programming

Autores

Lassi Haaranen\*, Petri Ihantola†, Juha Sorva\*, Arto Vihavainen‡  
\*Aalto University, Department of Computer Science  
†Tampere University of Technology, Department of Pervasive Computing  
‡University of Helsinki, Department of Computer Science

Resumen

**Abstract**—A small number of recent studies have suggested that learning is enhanced when the illustrations in instructional materials are designed to appeal to the learners' emotions through the use of color and the personification of key elements. We sought to replicate this *emotional design effect* in the context of introductory object-oriented programming (OOP). In this preliminary study, a group of freshmen studied a text on objects which was illustrated using anthropomorphic graphics while a control group had access to abstract graphics. We found no significant difference in the groups' scores on a comprehension post-test, but the experimental group spent substantially less time on the task than the control group. Among those participants who had no prior programming experience, the materials inspired by emotional design were perceived as less intelligible and appealing and led to lower self-reported concentration levels. Although this result does not match the pattern of results from earlier studies, it shows that the choice of illustrations in learning materials matters and calls for more research that addresses the limitations of this preliminary study.

This paper reports a between-subjects experiment through which we explored the applicability of emotional design to learning materials for programming. We sought to determine whether a form of emotional design improves students' learning of introductory-level object-oriented programming and whether it affects the students' assessments of the affective and cognitive aspects of a learning situation in this context. More generally, our work seeks to gauge the robustness of the emotional design effect.

The present study is a partial replication of earlier experiments on the impact of emotional design in multimedia instruction, albeit within the novel domain of computing and using a particular implementation of emotional design. We provide a review of the related work in Section II. Section III discusses the methodology of our study. The results are discussed in Section V. We conclude the article in Section VI by considering the limitations of our study and the future work that could address them.

## Trabajo Relacionado

Introducción

### I. INTRODUCTION

An everyday object such as a wine glass works better if its design brings about positive emotions in us [1]. What is more, just as objects can be designed to influence the emotions of those who see and use them, so can learning situations and materials. This is the basis of *emotional design*, an approach to multimedia instruction in which instructional materials are designed to engage learners emotionally. In the past few years, a small body of evidence has emerged in support of the idea that meaningful improvements on learners' emotional states and the effectiveness of their learning might be achieved with something as simple as the judicious use of color and the personification of visualizations by making key elements appear more human-like.

As students of computing increasingly use electronic textbooks and massively shared multimedia learning materials, and as there is an ever-present need to learn a lot within a limited time and with limited human tutoring available, it is important to explore any potential advance in promoting time-effective learning. If the emotional design effect is real and general, it is easy to find plausible use cases for it within computing. For instance, when teaching the basics of object-oriented programming, the emotional design effect would imply that learning is boosted by the use of expressive anthropomorphic color graphics that illustrate key interactions between the objects in a program, rather than abstract diagrams.

### II. RELATED WORK

The first subsection below describes prior studies of emotional design in multimedia learning. The second reviews the work within computing education research that most closely links to emotional design, and provides a short commentary on it from an emotional design perspective.

#### A. Emotional Design of Multimedia

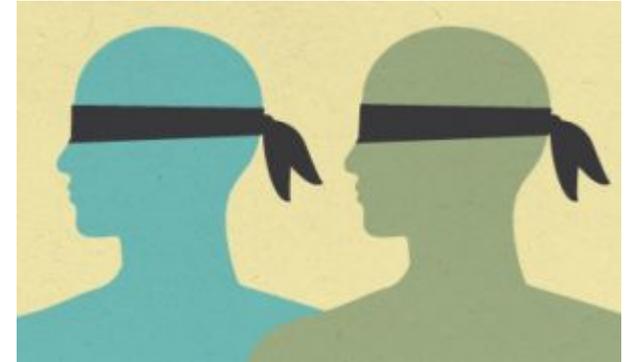
As a design principle for multimedia instruction, emotional design means "redesigning the graphics in a multimedia lesson to enhance the level of *personification* and *visual appeal* of the *essential elements* in the lesson" [2] (emphases added). It differs from merely decorating a lesson with interesting images in that the visuals are designed to highlight educationally critical elements and their appearance is chosen so that it may engage learners' emotions either consciously or unconsciously.

In this sense of the term, emotional design was first explored by Um et al. in 2012 [3]. They applied emotional design to a multimedia lesson for adult learners on the human immune system by giving colorful, expressive faces to key elements such as cells. Students in the treatment group scored higher on a post-tests of comprehension and transfer and reported lower perceived difficulty than those in a control group with unpersonified grayscale graphics. In two follow-up studies by

# Tipos de publicaciones

---

- Literatura científica con referato
  - con revisión de pares, ciego o doble ciego
  - en general indexado
- Literatura gris
  - no indexado o sin referato
  - tesis de maestría, tesis de doctorado, reportes técnicos
- Divulgación científica
  - publicaciones que interpretan y hacen accesible el conocimiento científico a la sociedad
- Artículos de opinión
  - revistas, foros, stackoverflow, etc.



# Acceso a publicaciones científicas

---

- Los artículos científicos se pueden acceder comprando las revistas, yendo a las conferencias o mediante motores de búsqueda de literatura o librerías digitales. Estos últimos indexan material de otras fuentes.
- En Uruguay contamos con el timbó (<http://www.timbo.org.uy/>)



- Otros medios
  - libgen → <https://libgen.is/>
  - sci.hub → <https://sci-hub.ru/>
  - z-lib → <https://singlelogin.re/>

# Publicaciones - Ejercicio en grupo

---

- Conteste para cada publicación las siguientes preguntas:
  - ¿Qué tipo de publicación es?
  - ¿Cuál es el propósito del artículo?
  - ¿Cuál es el método de investigación utilizado?

# Próxima clase - ¿Qué tengo que hacer?

---

- Formar grupos para todo el trabajo del curso.
- Traer ideas de tema de investigación.
- Hacer un usuario de Timbó y probar la plataforma.
- Leer capítulos 1, 2 y 3 del libro.