



JULIO ANTHONY ENGELS RUIZ COTO – 1284719

# Análisis y diseño II

## Evaluación Parcial I

### Fundamentos teóricos

1. Contestar el cuestionario disponible en el portal académico de la universidad.
2. Una vez completado el cuestionario, resolver los retos que se presentan.

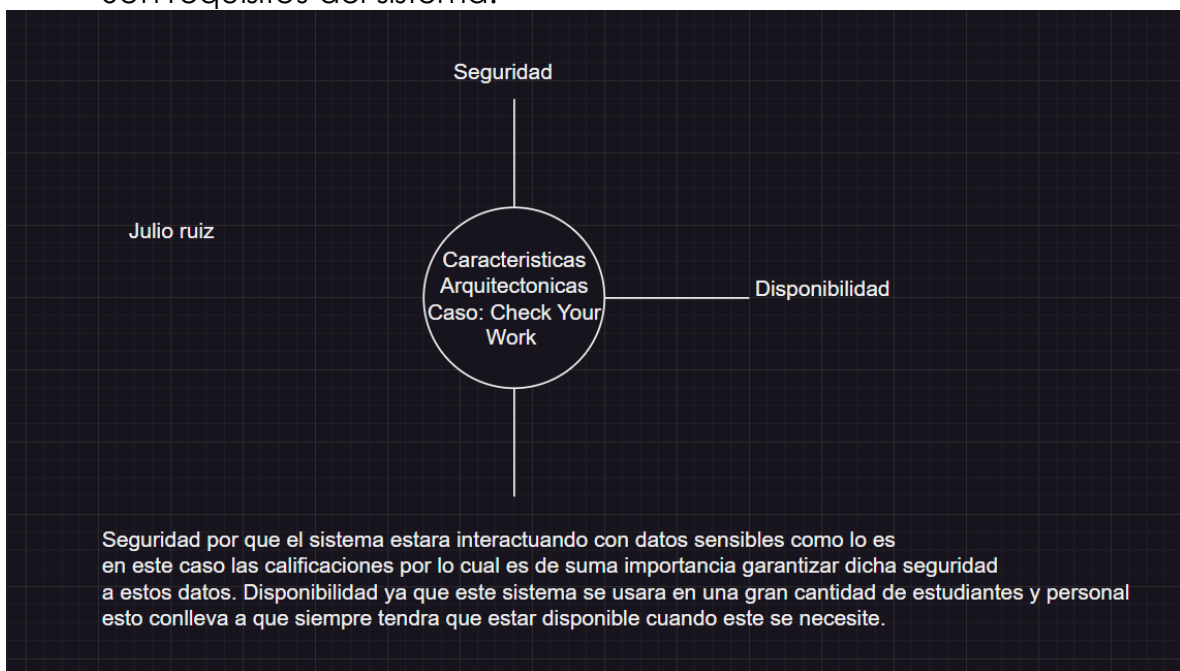
### Reto I

#### Diseño de arquitectura a partir del análisis

Instrucciones: dado el siguiente caso, diseñar la arquitectura de software considerando los puntos a continuación.

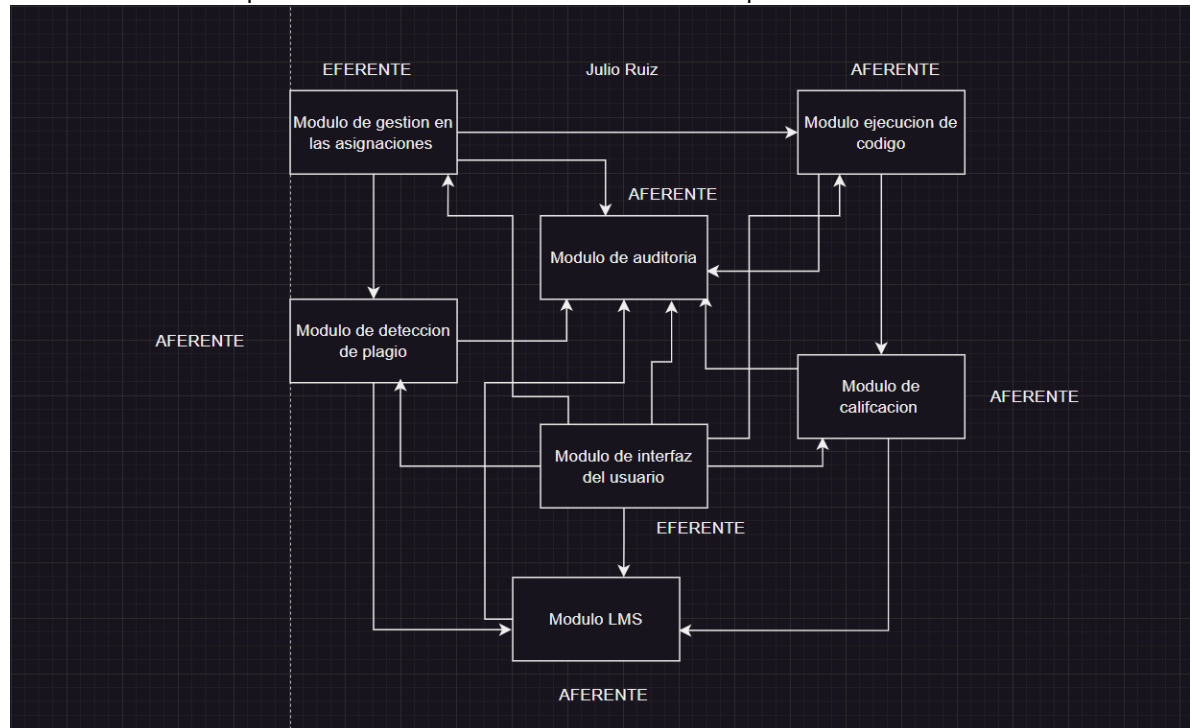
1. Elección de las características claves. Utilice al menos dos métodos<sup>1</sup> e identifique qué tipo de características utilizó y por qué.

Son requisitos del sistema:



<sup>1</sup> Pueden usar las plantillas proporcionadas en el curso.

## 2. Diseño de componentes e identificación de dependencias.



## 3. Definir el rol del equipo de arquitectos, desde el pensamiento arquitectónico y su relación con los desarrolladores (diferencias con el pensamiento de diseño).

El rol del equipo de arquitectos conlleva a una importante evaluación y manejo de los riesgos técnicos y arquitectónicos con esto ayudan a prever problemas y prepararse para estos mismos a que ocurran, los arquitectos necesitan pensar en términos de interfaces, capas y componentes, también necesitan considerar aspectos como la seguridad, escalabilidad, etc. Los arquitectos deben estar dispuestos a poder escuchar y considerar la retroalimentación de los desarrolladores, los arquitectos encargados tienen la visión global del sistema en el que se encuentren trabajando, por otro lado los desarrolladores tienen un conocimiento mas profundo de estos detalles técnicos y esto ofrece una información muy valiosa.



### Check Your Work

A university has greatly expanded its CS course and wants to be able to automate the grading of simple programming assignments.

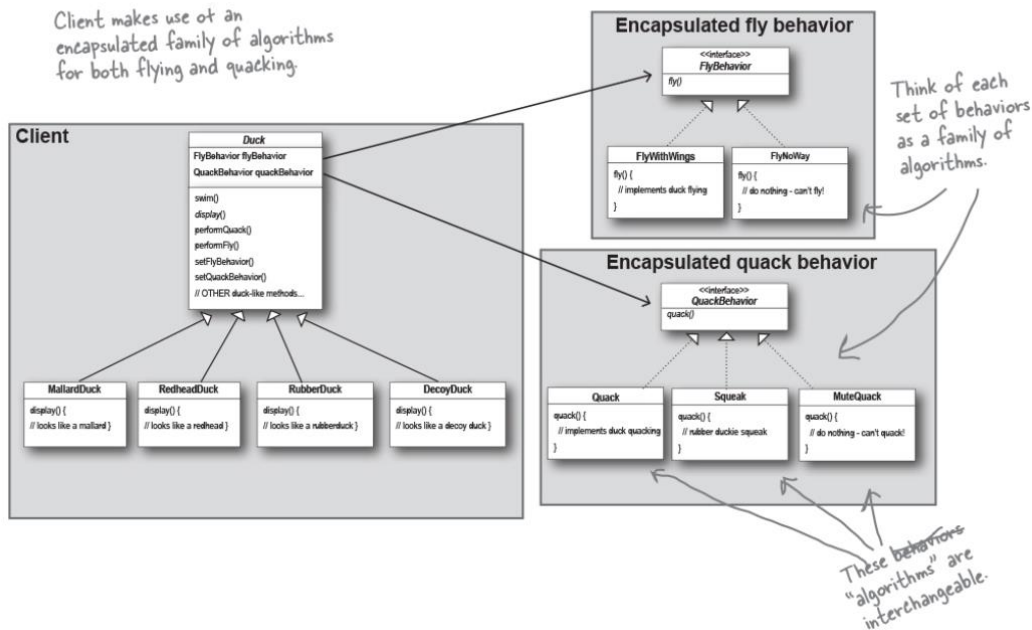
- Users: 300+ students per year, plus staff and admin.
- Requirements:
  - students must be able to upload their source code, which will be run and graded
  - grades and runs must be persistent and audit-able
  - required plagiarism detection system involving comparing with other submissions and also submitting to a web-based service (TurnItIn).
  - integration required with the University's learning management system (LMS)
  - professor sets due date and time, after which submissions are rejected
  - students can submit as many attempts as they want to improve their grade
  - professors determine grading criteria, which may include metrics and/or tests
- Additional Context:
  - University's LMS is mainframe based and quite difficult to make changes to
  - grades are audited each year by state-based regulatory body
  - University has very little budget for IT as it is building a spare stadium for SportsBall
  - University has a record for highest-performing CS graduates in the country

## Reto II

### Análisis de patrones de diseño

Instrucciones: dado el patrón estrategia, conteste las siguientes preguntas

#### Head First Design Patterns: Building Extensible and...



1. ¿Qué tipo de patrón es de arquitectura o de diseño; y por qué?  
*R// Este es un patrón de diseño tiene la intención de definir una familia de algoritmos y encapsular cada uno de ellos y hacerlo intercambiables, más que todo se utiliza para separar los algoritmos de los objetos que los usan.*
2. Identifique dos principios de diseño que ofrece el patrón.  
*R// Este patrón tiene como primer principio, el principio de encapsulación: este patrón permite el encapsulamiento de comportamiento de sus propios objetos lo que puede simplificar el mantenimiento a medida que el sistema escale. El segundo principio que tiene es, el principio de programar para interfaces, esto permite cambiar las estrategias en tiempo de ejecución sin cambiar el contexto donde se estén utilizando.*
3. ¿Qué tipo y nivel de acoplamiento se maneja? Explique y complemente con ejemplos de código.  
*R// Este patrón tiene un acoplamiento de tipo llamado loose coupling, por que los algoritmos son independientes de los objetos que utiliza, dicho en otras palabras, los clientes y las estrategias no están estrechamente acoplados. El patrón reduce el acoplamiento mínimo lo que resulta con una alta cohesión quiere decir que cada componente se centra en lo que se suponen que tiene que hacer.*

4. Configure estas clases según el patrón estrategia y coloque el método `setWeapon` en la clase correcta.

