**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

**FACULTAD DE INGENIERÍA**

**SISTEMA DE REGISTRO Y CONTROL DE PROCESO DE ADMISIÓN**

**PROYECTO PROFESIONAL PRESENTADO POR GRUPO N° ¿:**

* **Yvan Román Algorta (U201925624)**
* **Andres Inope Sandoval (U202017537)**
* **Alex de la cruz blas (U202012657)**
* **Bryam Soto Montes (U202016391)**

**PARA EL CURSO DE PROGRAMACIÓN ORIENTADA A OBJETOS - IS210 - E23A**

**PROFESOR:**

**ING. Carlos Alberto Flores Orihuela**

**Lima, 02 de mayo de 2020**

# RESUMEN

El presente trabajo se basa en realizar un programa orientado a objetos para la gestión de un concurso que tiene alta demanda para ingresar a los colegios de alto rendimiento. Este concurso está organizado por el Ministerio de Educación.

Para ello, emplearemos como lenguaje de programación Ruby y aplicaremos los conceptos de clases y sus relaciones entre ellas, objetos, herencia, polimorfismo y patrones de diseño.

A continuación, nombraremos algunas ventajas del por qué utilizar programación orientado a objectos para este trabajo:

* Fomenta la reutilización y extensión del código
* Permite crear sistemas más complejos
* Relacionar el sistema al mundo real
* Facilita la creación de programas visuales
* Agiliza el desarrollo de software
* Facilita el mantenimiento del software

**Índice:**

[1. RESUMEN 1](#_Toc39249504)

[2. INTRODUCCIÓN 3](#_Toc39249505)

[3. ENUNCIADO DEL PROYECTO: 3](#_Toc39249506)

[4. DIAGRAMA DE CLASES INCLUYENDO PATRONES DE DISEÑO 7](#_Toc39249507)

[5. URL DEL REPOSITORIO DE PROGRAMA FUENTE 8](#_Toc39249508)

[6. RESULTADO DE PRUEBAS UNITARIAS 8](#_Toc39249509)

[7. CONCLUSIONES 11](#_Toc39249510)

[8. RECOMENDACIONES 11](#_Toc39249511)

[9. GLOSARIO 11](#_Toc39249512)

[10. EVIDENCIAS DE TRABAJO EN EQUIPO 12](#_Toc39249513)

[11. BIBLIOGRAFIA 14](#_Toc39249514)

[12. OBJETIVO DEL ESTUDIANTE (STUDENT OUTCOME) 15](#_Toc39249515)

# INTRODUCCIÓN

El ministerio de Educación para este concurso que tiene alta demanda, nos solicitó un sistema de registro y control de proceso de admisión para el ingreso a los colegios de alto rendimiento que hay en el Perú, para esto se está considerando evaluar el ingreso de 3 factores:

* Calificación Socioeconómica.
* Rendimiento en el 2do grado.
* Evaluación de conocimiento.

Para ello, el sistema debe permitir configurar e ingresar las respuestas de los exámenes para que la calificación sea de forma automática y se tenga una calificación en base a 100. Por cada pregunta incorrecta, el sistema restará la mitad del puntaje, en el caso del tipo de evaluación de 10 preguntas se descontará 5 puntos. Y si el alumno no contestó una pregunta no se suma ni se resta puntaje. Por otro lado, el sistema debe permitir ingresar la cantidad de vacantes. De esta configuración dependerá el estado del alumno: SI INGRESA o NO INGRESA, ordenandos por el puntaje final que considere los 3 factores mencionados anteriormente.

También debe mostrar resultados de manera estadísticas, según lo requerido por el ministerio de Educación.

# ENUNCIADO DEL PROYECTO:

El Ministerio de Educación para el próximo año 2021 implementará una nueva política para el ingreso a los colegios de alto rendimiento, se considera evaluar el ingreso en base a 3 factores:

1. Calificación socioeconómica (CS)
2. Rendimiento en el 2do grado (RE)
3. Evaluación de conocimiento (EC)

Por la gran demanda de este concurso es necesario contar con un sistema informático que brinde apoyo en el tema de registro y control del proceso de admisión, razón por la cual solicitan a Ud. Aplicar sus habilidades y conocimientos de ingeniería para desarrollar este aplicativo.

Para el registro de los alumnos considerar:

* DNI
* Apellidos
* Nombres
* Edad
* Género

Si proviene de colegio nacional considerar:

* Colegio pertenece a rural o urbano
* Promedio ponderado del 2 año

Si proviene de colegio particular considerar:

* Monto de la pensión.
* Puesto en el que finalizó el 2do grado

Para todos los alumnos se debe registrar por lo menos un tutor (máximo 2) para quien se considera los siguientes datos:

* DNI
* Apellidos
* Nombre
* Parentesco

El sistema debe permitir configurar e ingresar las respuestas de los exámenes para que la calificación sea de forma automática. Para esto se debe registrar:

* Código de evaluación.
* Cantidad de preguntas, considerar 2 tipos: de 10 y de 20 preguntas.

**Reglas de Negocio:**

* El puntaje final para el alumno se obtiene de la siguiente manera:

20%(CS) + 30%(RE) + 50%(EC)

* Para obtener la calificación socioeconómica (CS) se considera:
  + Para alumnos de colegios nacionales:

|  |  |
| --- | --- |
| Zona | Puntaje |
| Rural | 100 puntos |
| Urbana | 80 puntos |

* + Para alumnos de colegios particulares:

|  |  |
| --- | --- |
| Monto de la pensión | Puntaje |
| <= 200 soles | 90 puntos |
| > 200 y <= 400 | 70 puntos |
| > 400 y <= 600 | 50 puntos |
| > 600 | 40 puntos |

* Para obtener el puntaje por Rendimiento en el 2do grado (RE)
  + Para alumnos de colegios nacionales:

|  |  |
| --- | --- |
| Promedio Ponderado | Puntaje |
| >= 19 | 100 puntos |
| >= 18 y < 19 | 80 puntos |
| >= 16 y < 18 | 60 puntos |
| >= 14 y < 16 | 40 puntos |
| >= 11 y < 14 | 20 puntos |
| < 11 | 0 puntos |

* + Para alumnos de colegios particulares:

|  |  |
| --- | --- |
| Puesto | Puntaje |
| Entre los 3 primeros puestos | 100 puntos |
| 5to puesto – 4to puesto | 80 puntos |
| 10mo puesto – 6to puesto | 60 puntos |
| 20vo puesto – 11vo puesto | 40 puntos |
| Menos de 20vo puesto | 0 puntos |

* De la configuración de la evaluación: El sistema solicitará ingresar las respuestas correctas a, b, c, d, e para la alternativa correcta. Dependiendo del tipo de evaluación (10 o 20 preguntas)
* Para rendir la evaluación el sistema contará con un simulador que podrá lanzarse cada vez que el usuario lo desee: Para todos los alumnos registrados y según la evaluación aplicada generará de manera aleatoria respuestas a las preguntas (a – b - c - d – e). El sistema cotejará estas respuestas aleatorias con las respuestas ingresadas y registrará el total de preguntas respondidas correctamente, la cantidad de preguntas incorrectas y también se debe considerar en la simulación una probabilidad de 0.1 que el alumno deje la pregunta sin responder.

De acuerdo a la cantidad de preguntas (10 o 20) el sistema realizará el cálculo para obtener el puntaje en base 100.

* Para el peso de los puntajes considerar:
  + Por cada pregunta contestada correcta se asignará un puntaje dependiendo del tipo de evaluación. Por ejemplo, si la evaluación consta de 10 preguntas. Cada pregunta correcta tendrá un puntaje de 10 puntos. Por cada pregunta incorrecta el sistema restará la mitad del puntaje, en el caso del tipo de evaluación de 10 preguntas se descontará 5 puntos. Y si el alumno no contestó una pregunta no se suma ni se resta puntaje.
* El sistema debe permitir ingresar la cantidad de vacantes. De esta configuración dependerá el estado del alumno: Si INGRESA o NO INGRESA, ordenados por el puntaje final que considera los 3 factores.

Reportes solicitados:

* Con el DNI del estudiante se imprimirá todos los datos del estudiante. Si la evaluación ya fue ejecutada debe mostrar el resultado detallado de todas las preguntas, su puntaje final y su estado: INGRESA o NO INGRESA.
* También es muy importante consultar según el DNI del alumno el o los tutores.
* Listar el resultado de la evaluación:
  + Todos los alumnos ordenados por puntaje final y estado.
  + Todos los alumnos ingresantes.
  + Todos los alumnos no ingresantes.
* Como parte estadística es importante mostrar:
  + Cantidad de alumnos postulantes masculinos y femeninos.
  + Ingresantes masculinos y femeninos
  + No ingresantes masculinos y femeninos
  + Ingresantes porcentaje de colegios nacionales y particulares.
  + No ingresantes porcentaje de colegios nacionales y particulares.
  + La edad de los estudiantes se encuentra entre 11 – 15 años, mostrar reporte cantidad de ingresantes y no ingresante por edades.

# DIAGRAMA DE CLASES INCLUYENDO PATRONES DE DISEÑO

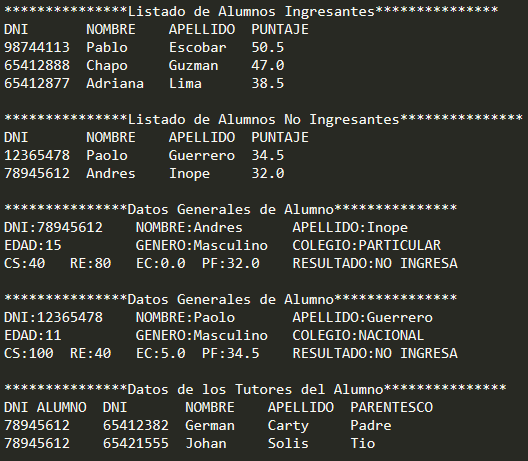
# URL DEL REPOSITORIO DE PROGRAMA FUENTE

<https://github.com/andresinope77/Coronamigos_Trabajo_Final>

# RESULTADO DE PRUEBAS UNITARIAS

Imagen que contiene texto, tabla

Descripción generada automáticamente

****

**Imagen que contiene tabla, negro, blanco, teléfono

Descripción generada automáticamente**

# CONCLUSIONES

* Este trabajo se ha basado en la realización de un programa orientado a objetos para poder gestionar correctamente un programa que se basa en un concurso que tiene alta demanda este nos permite ingresar a los colegios de alto rendimiento, donde se registrara y controlara el proceso de admisión aplicando ciertos factores que se reflejan en el programa.

# RECOMENDACIONES

* Este programa nos sirve de mucha ayuda cuando se utilice en otros colegios o instituciones que optaran usarlo en su proceso de admisión y tenga similares factores.
* El programa Ruby en donde se ha escrito el código nos permitió poder entender diversos casos en los cuales se ve reflejado el proyecto realizado.
* EL uso del git hub nos ayuda a plasmar el código que realizo el grupo este permite alojar proyectos utilizando el sistema de control de versiones Git.

# GLOSARIO

**HERENCIA:**

La herencia es un mecanismo de reutilización de código en la Programación Orientada por Objetos. A través de la herencia, **una clase hereda de otra sus atributos y métodos**. Técnicamente, los atributos no son heredados, se heredan los métodos que permiten leerlos o escribirlos (que se crean usando attr\_accessor, attr\_reader, y attr\_writer).

**POLIMORFISMO:**

Formalmente se define como la capacidad de un objeto de tomar otra(s) forma(s).

Otros lenguajes de programación como Java y C++ utilizan un concepto llamado interfaces para definir el contrato (o los métodos que debe implementar una clase para cumplir el contrato), pero la idea es la misma.

**CLASE Y OBJETOS:**

Las clases y los objetos son los conceptos más importantes de la Programación Orientada por Objetos, y están fuertemente relacionados. Los objetos se crean a partir de clases, y las clases definen los atributos y el comportamiento que tendrán los objetos. A los objetos también se les llama instancias de clase.

**FACTORIAL:**

Podemos calcular la factorial fácilmente tanto con un bucle como con una función recursiva.

**PATRONES DE DISEÑO:**

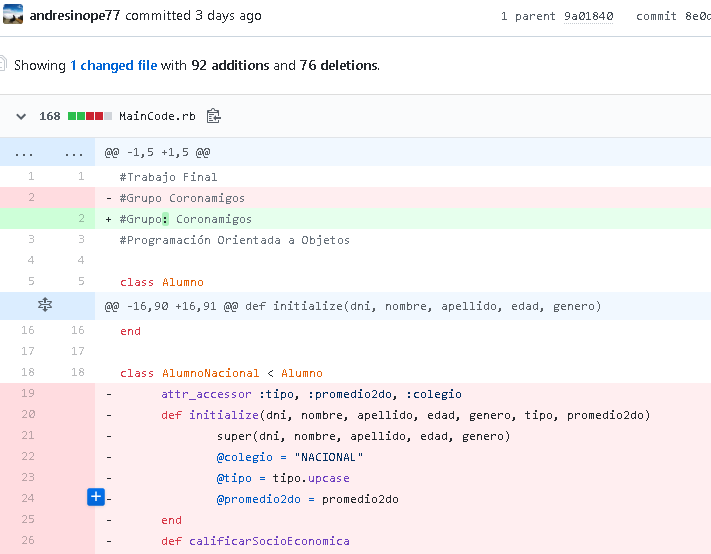
Un patrón de diseño es una solución general reutilizable a un problema de común ocurrencia en el diseño del software.

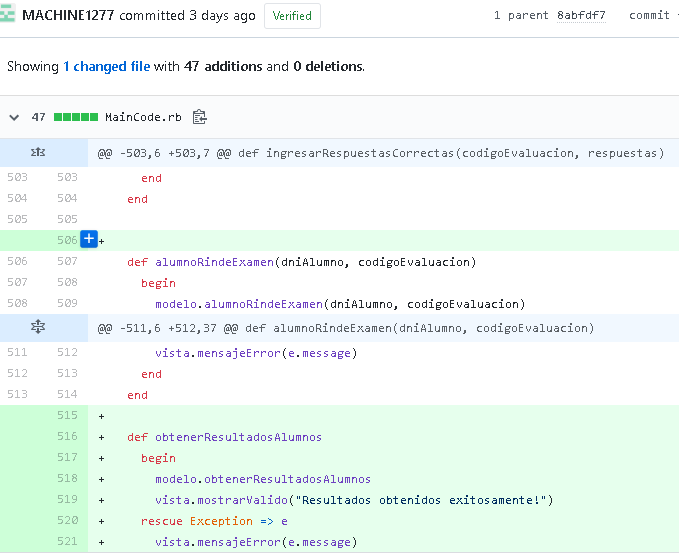
**DIAGRAMA DE CLASES:**

Los diagramas de clases son uno de los tipos de diagramas más útiles en UML, ya que trazan claramente la estructura de un sistema concreto al modelar sus clases, atributos, operaciones y relaciones entre objetos. Con nuestro [software de generación de diagramas UML](https://www.lucidchart.com/pages/examples/uml_diagram_tool), la creación de estos diagramas no es tan abrumadora como podría parecer.

# EVIDENCIAS DE TRABAJO EN EQUIPO

Trabajos realizados en la plataforma de github:





Captura de pantalla de un celular

Descripción generada automáticamente

Captura de pantalla de un celular

Descripción generada automáticamente

Registro de Trabajo de Github:

Una captura de pantalla de una red social

Descripción generada automáticamente

# BIBLIOGRAFIA

* Guías de patrones de diseño en Ruby <https://github.com/Elolawyn/RubyDesignPatterns>

# OBJETIVO DEL ESTUDIANTE (STUDENT OUTCOME)

Con respecto al desarrollo del curso se concluye que el trabajo en equipo y el empeño que uno da con sus aportaciones para un fin común, es una de las partes más importantes, porque se define los diferentes tipos de tareas a realizar. Según lo aprendido en el transcurso del curso o a base de las mismas experiencias de los compañeros, que uno llega asimilar lo aprendido en aula y como también de una nueva metodología que son clases virtuales. Utilizando como herramientas de trabajo como el StarUML, Sublime Text aprendido en las clases presenciales y no presencial con el uso de Github que es una plataforma que nos ayuda a interactuar a todos nosotros a base de un repositorio creado, que el grupo puede reutilizar y actualizar en cualquier momento del día. Además, su aprendizaje constituye una oportunidad al mejorar el razonamiento lógico formal.