

# **UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

**FACULTAD DE INGENIERÍA**

**DIVISIÓN DE ESTUDIOS PROFESIONALES PARA EJECUTIVOS**

**PROGRAMACION ORIENTADA A OBJETOS**

**INTEGRANTES:**

**GRUPO 1**

* Anton Ramirez, Nicolas U202010885
* Arrieta Jeri, Diana U201924447
* Bello Gutierrez, Francisco U202920389
* Medina Valencia, Frank U20181B718
* Mendoza Paredes, Brayan U201915265
* Mendoza Flores, Paolo U202020043

**PROFESOR:**

Carlos Flores Orihuela

**Lima, noviembre del 2020**

# **RESUMEN**

Para el desarrollo del presente trabajo se ha elegido a la Universidad Peruana de ciencias Aplicadas (UPC) que es una Universidad dedicada principalmente a la formación y capacitación de personas en las diferentes áreas Ingeniería, Gestión, Negocios, Diseño y comunicaciones. UPC inició sus actividades educativas hace 25 años y actualmente cuenta con 4 Sedes en diversos distritos de Lima Metropolitana.

**ÍNDICE**

1. [**RESUMEN 2**](#_Toc58029461)
2. [**INTRODUCCION 4**](#_Toc58029462)
3. [**ENUNCIADO 5**](#_Toc58029463)
4. [**REGLAS DE NEGOCIO: 6**](#_Toc58029464)
5. [**CRONOGRAMA DE ACTIVIDADES 10**](#_Toc58029465)
6. [**Plan calendario de Reuniones: 10**](#_Toc58029466)

[**Detalle de Reuniones: 11**](#_Toc58029467)

[**Detalle de Actividades: 12**](#_Toc58029468)

1. [**DIAGRAMA DE CLASES DEL MODELO 13**](#_Toc58029469)
2. [**DIAGRAMA DE PATRONES DE DISEÑO 14**](#_Toc58029470)
3. [**ENLACE DE VIDEO DE PRUEBAS 15**](#_Toc58029471)
4. [**RESULTADO PRUEBAS UNITARIAS 16**](#_Toc58029472)
5. [**GLOSARIO 21**](#_Toc58029473)
6. [**EVIDENCIAS DE TRABAJO EN EQUIPO 22**](#_Toc58029474)
7. [**CONCLUSIÓN 25**](#_Toc58029475)
8. [**RECOMENDACIONES 25**](#_Toc58029476)
9. [**BIBLIOGRÁFICA 26**](#_Toc58029477)

# **INTRODUCCION**

El presente trabajo consiste en el desarrollo y diseño de un programa Orientado a Objetos, dicho programa cubrirá una necesidad en el sector salud denominado “Solicitud de registro de los colaboradores de la universidad UPC para acceder a un seguro privado (EPS)”. El trabajo tiene como objetivo beneficiar a los trabajadores y la universidad mediante la automatización de registro para el EPS. Por lo tanto, en el proyecto se aplicará los métodos de clases, herencia y polimorfismo patrones y MVC aprendido en el curso.

A continuación, desarrollaremos el enunciado y pasos siguientes del proyecto.

# **ENUNCIADO**

La Universidad UPC requiere implementar un nuevo programa para gestionar los datos de sus colaboradores y permita evaluarlos para afiliarlos a un seguro privado (EPS). Los afiliados a la EPS pueden ser: Administrativos y Profesores a Tiempo Completo.

Todo Solicitante al EPS es registrado con:

* Solicitud de Afiliación
* Mes de afiliación
* Tipo de plan EPS
* Documento de identidad
* Código Colaborador
* Nombres y apellidos
* Edad
* Número de Dependientes (Hijos o Conyugue)
* Código de sede
* Descripción de sede
* Tipo de Contrato
* Tipo de fondo de pensión
* Seguro de invalidez

Si es un Solicitante Administrativo se debe considerar, además:

* Cargo
* Sueldo básico

Si es un Solicitante Profesor a Tiempo Completo se debe considerar también:

* Sueldo base
* Cantidad de años de antiguedad

# **REGLAS DE NEGOCIO:**

* La universidad UPC cuenta con 2 grupos de empleados: los administrativos y los docentes a tiempo completo.
* Cada solicitante que desee afiliarse al sistema EPS del instituto, debe contar con lo siguiente:

|  |
| --- |
| **Requisitos** |
| Solicitud de Afiliación |
| Mes de afiliación |
| Tipo de plan EPS |
| Documento de identidad |
| Código Colaborador |
| Nombres y apellidos |
| Edad |
| Número de Dependientes (Hijos o Conyugue) |
| Código de sede |
| Descripción de sede |
| Tipo de Contrato |
| Tipo de fondo de pensión |
| Seguro de invalidez |

* Al afiliar el titular, también deben afiliar a sus dependientes en caso de tenerlos, es decir al conyugue y/o hijos menores de 18 años.
* Para el caso del Solicitante Administrativo el atributo “Cargo” se puede considerar las siguientes posiciones:

|  |
| --- |
| **Cargo** |
| Gerente |
| Director |
| Jefe |
| Coordinador |
| Analista |
| Asesor |
| Tecnico |

* Para calcular las deducciones de los solicitantes

|  |  |
| --- | --- |
| Calculo de deducciones | |
| Deducciones | (Tarifa de EPS + Seguir de Invalidez + quinta Categoría + Fondo de pensión) |

* Para calcular el sueldo del Solicitante:

|  |  |
| --- | --- |
| ADMINISTRATIVO | |
| Sueldo Total | (Sueldo Básico + bono edad +gratificación) - Deducciones |

|  |  |
| --- | --- |
| **Edad** | **Bono** |
| mayores a 35 | 15% |
| mayores a 55 | 25% |

|  |  |
| --- | --- |
| PROFESOR | |
| Sueldo Total | (Sueldo Base + bono antigüedad +gratificación) - Deducciones |

|  |  |
| --- | --- |
| **Años antigüedad** | **Bono** |
| desde 5 hasta 10 | S/ 1500 |
| desde 10 hasta 15 | S/ 2500 |
| desde 15 hasta 20 | S/ 3500 |
| Mayor o igual a 20 | S/ 4000 |

* Para calcular la tarifa EPS del solicitante se considera los siguientes planes tarifarios:

|  |  |  |  |
| --- | --- | --- | --- |
| Descuento EPS – por planilla UPC | | | |
| Composición Familiar | Plan Base | Adicional 1 | Adicional 2 |
| Titular solo | 31.00 | 51.00 | 71.00 |
| Titular + 1 dependiente | 62.00 | 102.00 | 142.00 |
| Titular + 2 dependiente | 93.00 | 153.00 | 213.00 |
| Titular + 3 dependiente a mas | 124.00 | 204.00 | 284.00 |
| Hijos mayores de 18 años | 73.00 | 93.30 | 113.00 |

* La universidad evalúa los datos requeridos y registra al solicitante.

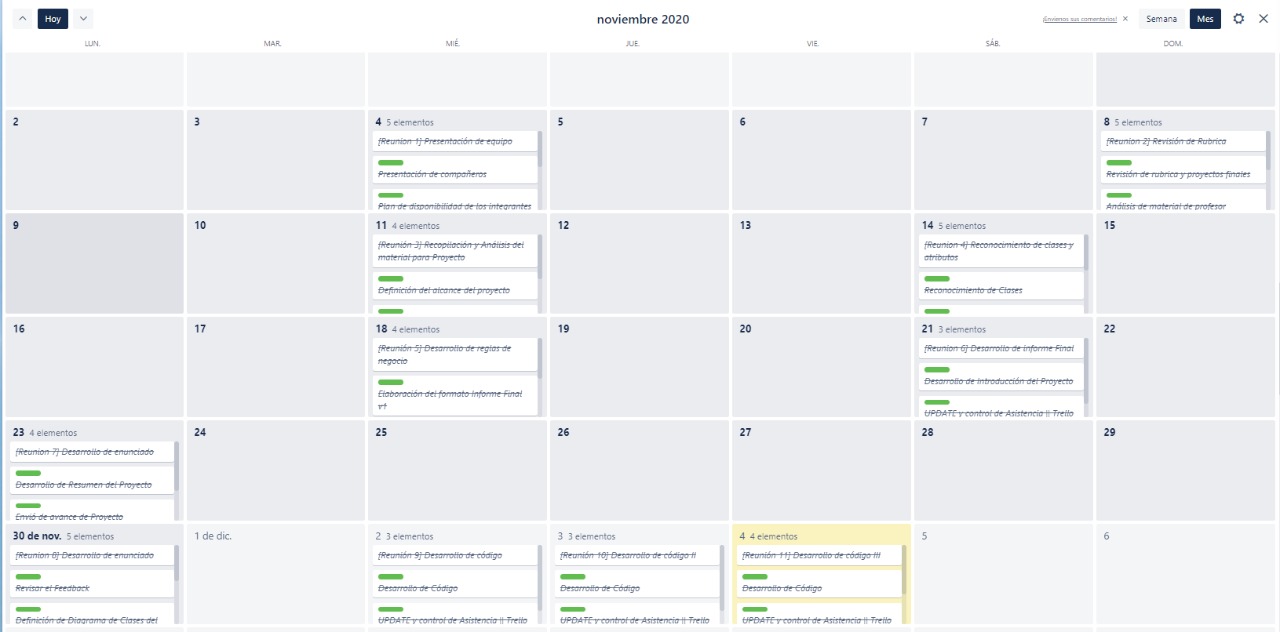
Reportes solicitados:

El programa debe realizar lo siguiente

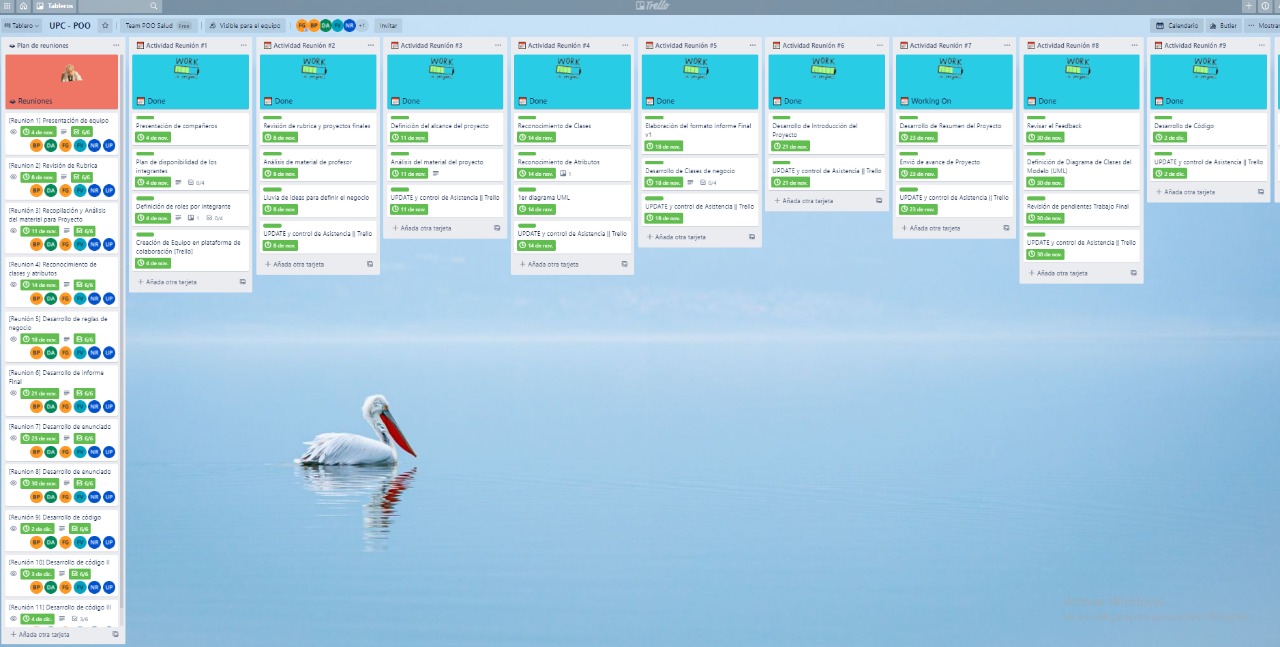
* Registrar a los solicitantes que cumplan con las condiciones requeridas para la afiliación EPS.
* Lista a todos a los solicitantes con código de solicitud
* Evaluar y listar registro de los solicitantes
* Lista a los afiliados con Dependientes
* Mostrar la cantidad de afiliados
* Calcula el Importe total del EPS
* Buscar afiliado ingresando el documento de identidad
* Obtener el solicitante con mayor sueldo
* Obtener el solicitante con menor sueldo
* Obtener el Administrativo con mayor sueldo
* Obtener el profesor con mayor sueldo

# **CRONOGRAMA DE ACTIVIDADES**

## **Plan calendario de Reuniones:**

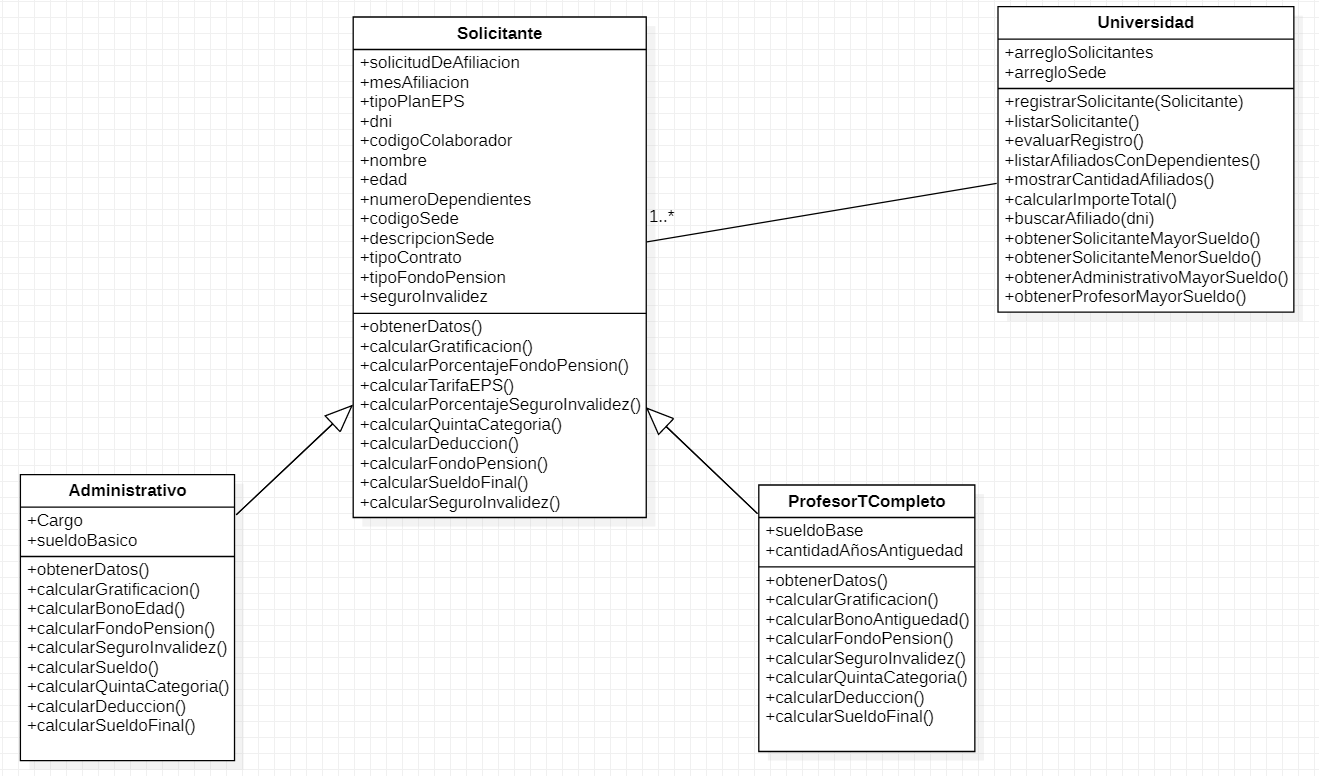


## **Detalle de Reuniones:**



## **Detalle de Actividades:**

## **DIAGRAMA DE CLASES DEL MODELO**



## **C:\Users\bmendoza\Downloads\WhatsApp Image 2020-12-05 at 02.15.21.jpegDIAGRAMA DE PATRONES DE DISEÑO**

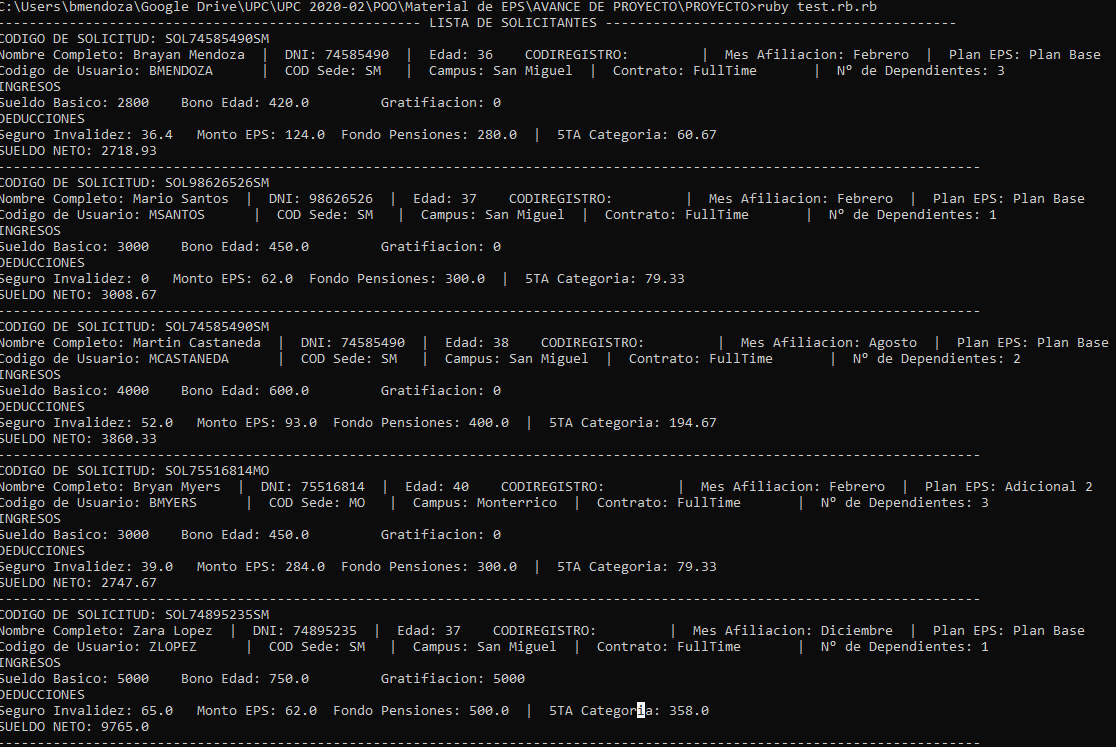
## **ENLACE DE VIDEO DE PRUEBAS**

Dar clic en el siguiente enlace para poder visualizar el video.

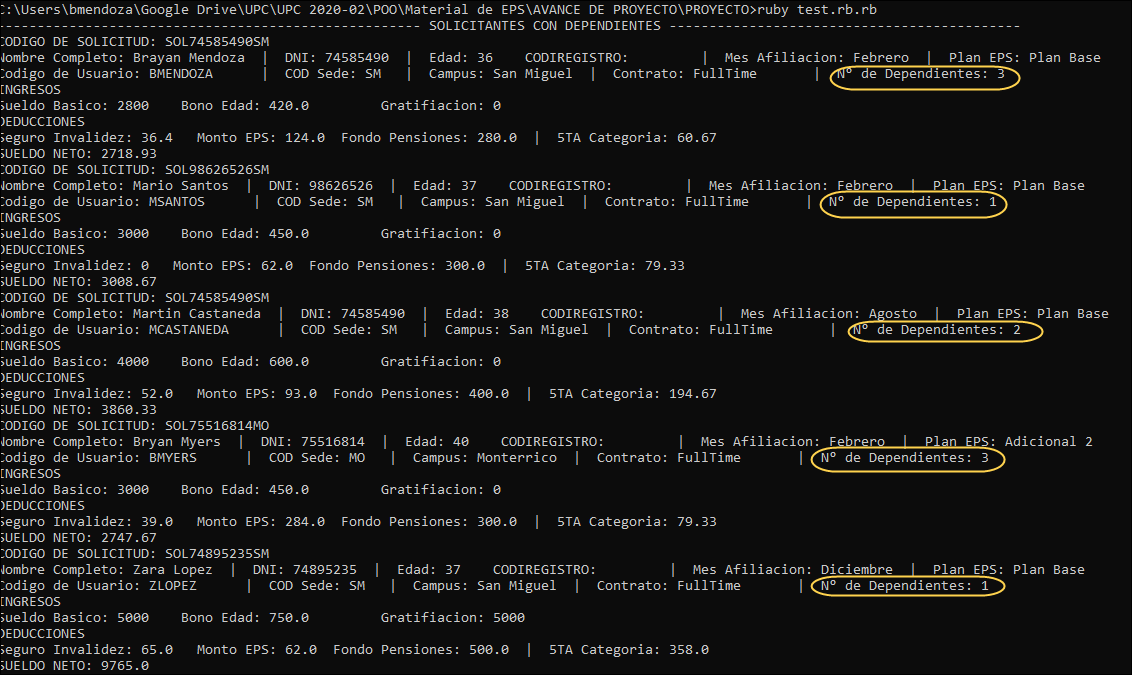
<https://bit.ly/36Ke1eR>

## **RESULTADO PRUEBAS UNITARIAS**

# LISTAR SOLICITANTES



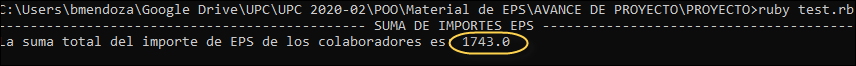
#LISTAR AFILIADOS CON DEPENDIENTES



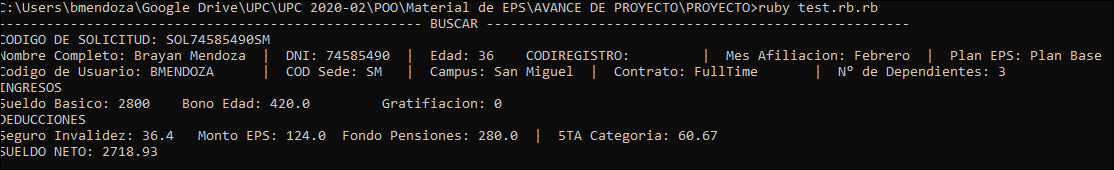
#MOSTRAR CANTIDAD DE AFILIADOS



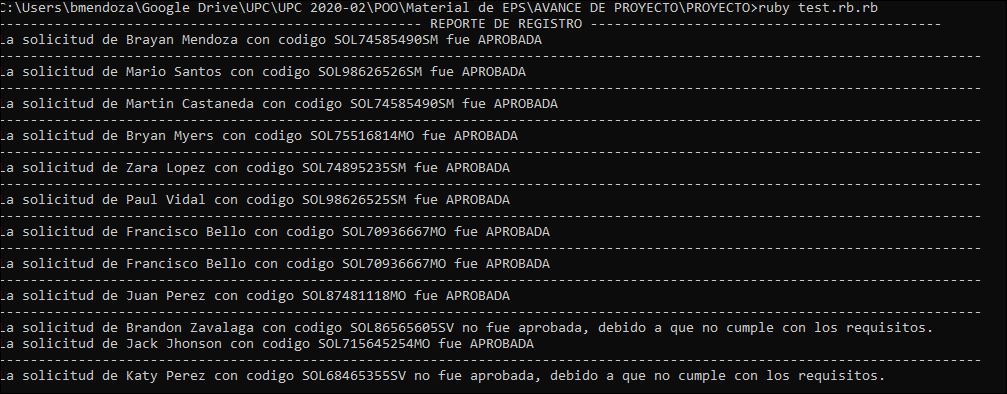
#MOSTRAR EL IMPORTE TOTAL ACMULADO DE EPS



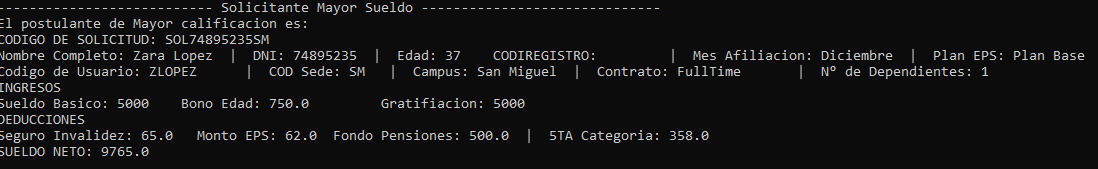
#BUSCAR SOLICITANTE MEDIANTE DNI



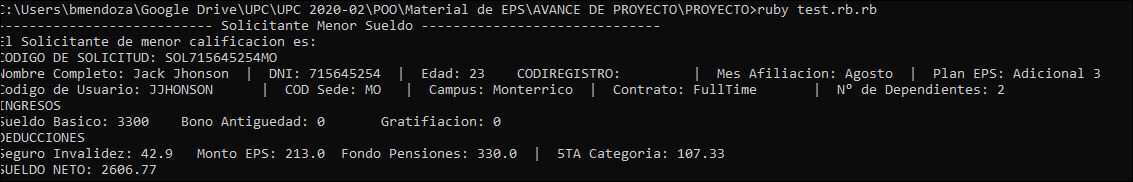
#EVALUAR SOLICITUD



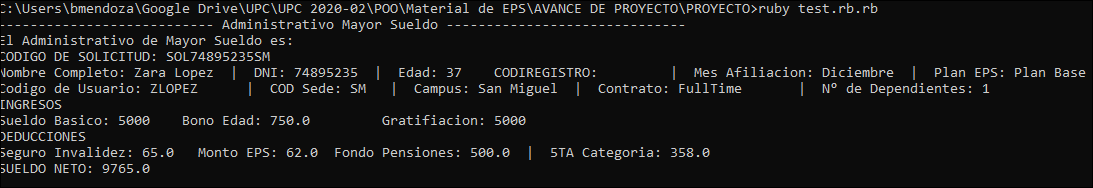
#MOSTRAR SOLICITANTE CON MAYOR SUELDO



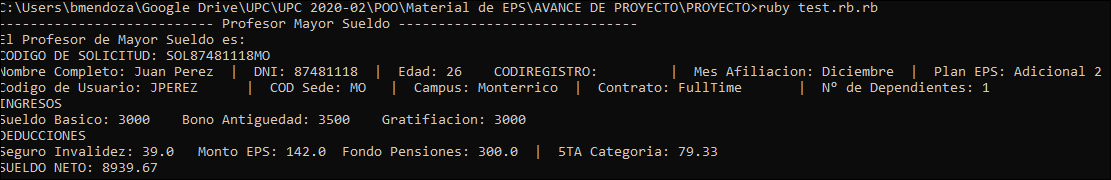
#MOSTRAR SOLICITANTE CON MENOR SUELDO



#MOSTRAR ADMINISTRATIVO MAYOR SUELDO



#MOSTRAR PROFESOR MAYOR SUELDO



## **GLOSARIO**

**Objeto**: Representan una entidad concreta o abstracta del mundo real, en programación básicamente se le conoce como la instancia de una clase en si es lo que da sentido a estas.

**Clase**: Son uno de los principales componentes de un lenguaje de programación, pues en ellas ocurren todos los procesos lógicos requeridos para un sistema.

**Atributos**: Son características específicas de una clase es donde se almacena la información de un objeto.

**Instancia**: Dar valores a los atributos de la clase para definir un objeto.

**Encapsulamiento**: Es uno de los más importantes en términos de seguridad dentro de nuestra aplicación, la encapsulación es la forma de proteger nuestros datos dentro del sistema, estableciendo básicamente los permisos o niveles de visibilidad o acceso de nuestros datos.

**Herencia**: Representa lo que conocemos de herencia en el mundo real, básicamente, mediante esta obtenemos las características o rasgos comunes de nuestros padres o abuelos, en java es el mismo enfoque permitiendo la creación de nuevas clases basadas en las clases ya existentes.

**Polimorfismo**: La capacidad que tienen los objetos de comportarse de múltiples formas sin olvidar que para esto se requiere de la herencia, en si consiste en hacer referencia a objetos descendientes.

**Método**: Representan el comportamiento u operaciones, la forma como interactúa la clase con su entorno.

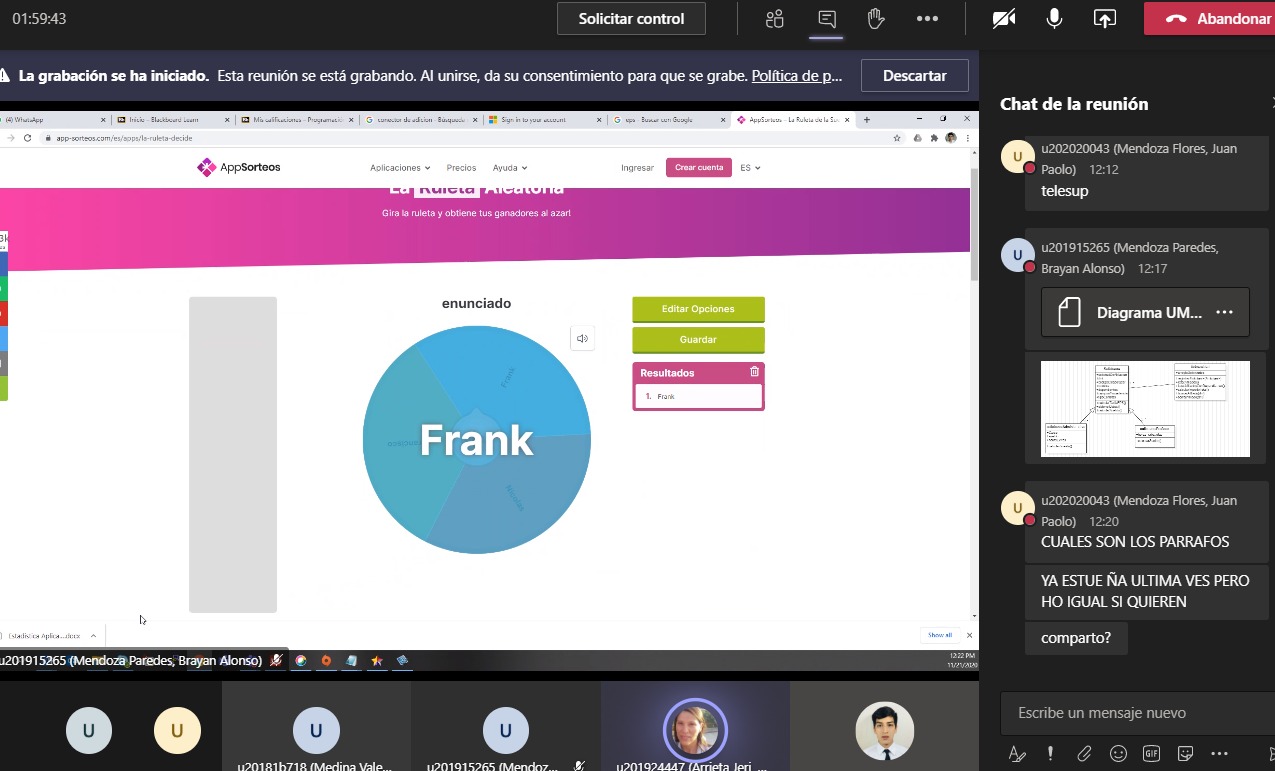
**Factory**: Es un patrón de diseño que permite la creación de objetos de un subtipo determinado a través de una clase Factory.

**MVC**: Es un patrón de diseño arquitectónico que permite separar las funciones del programa en modelos, vistas y controladores.

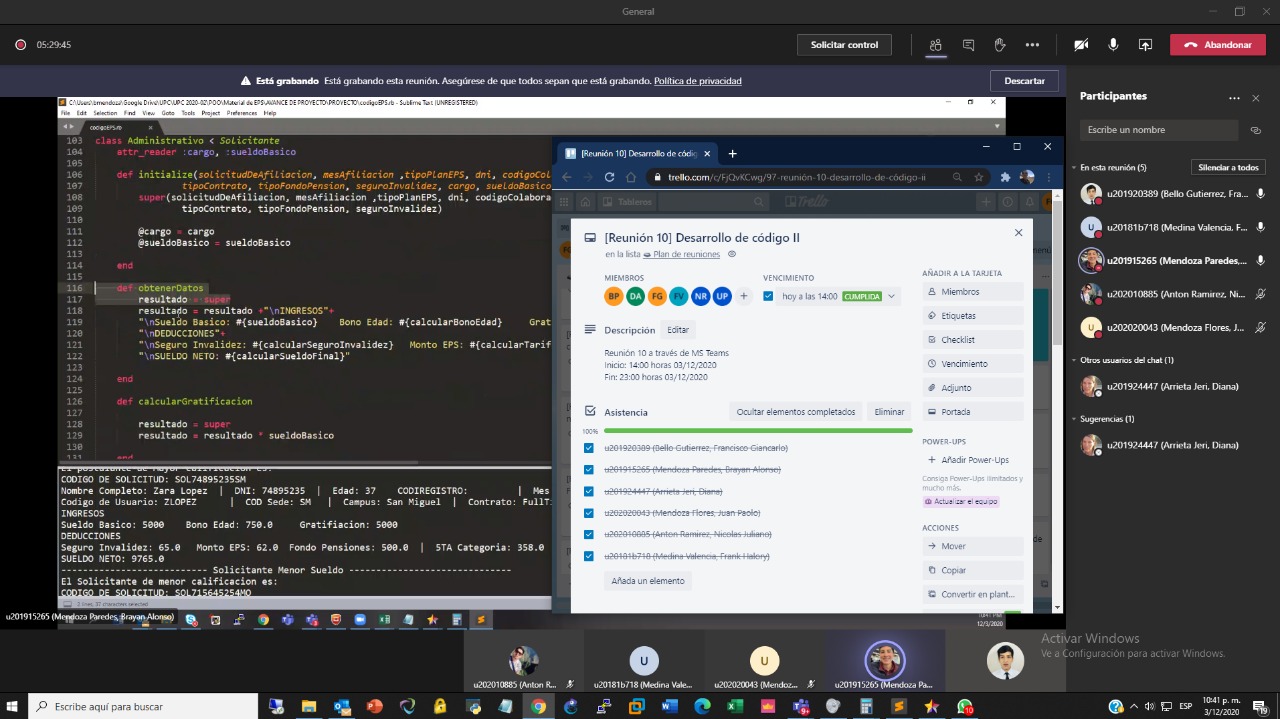
**Constructor**: Es una subrutina cuya misión es inicializar un objeto de una clase. En el constructor se asignan los valores iniciales de nuevo objeto. Se utiliza para crear tablas de clases virtuales y poder así desarrollar el polimorfismo.

## **EVIDENCIAS DE TRABAJO EN EQUIPO**

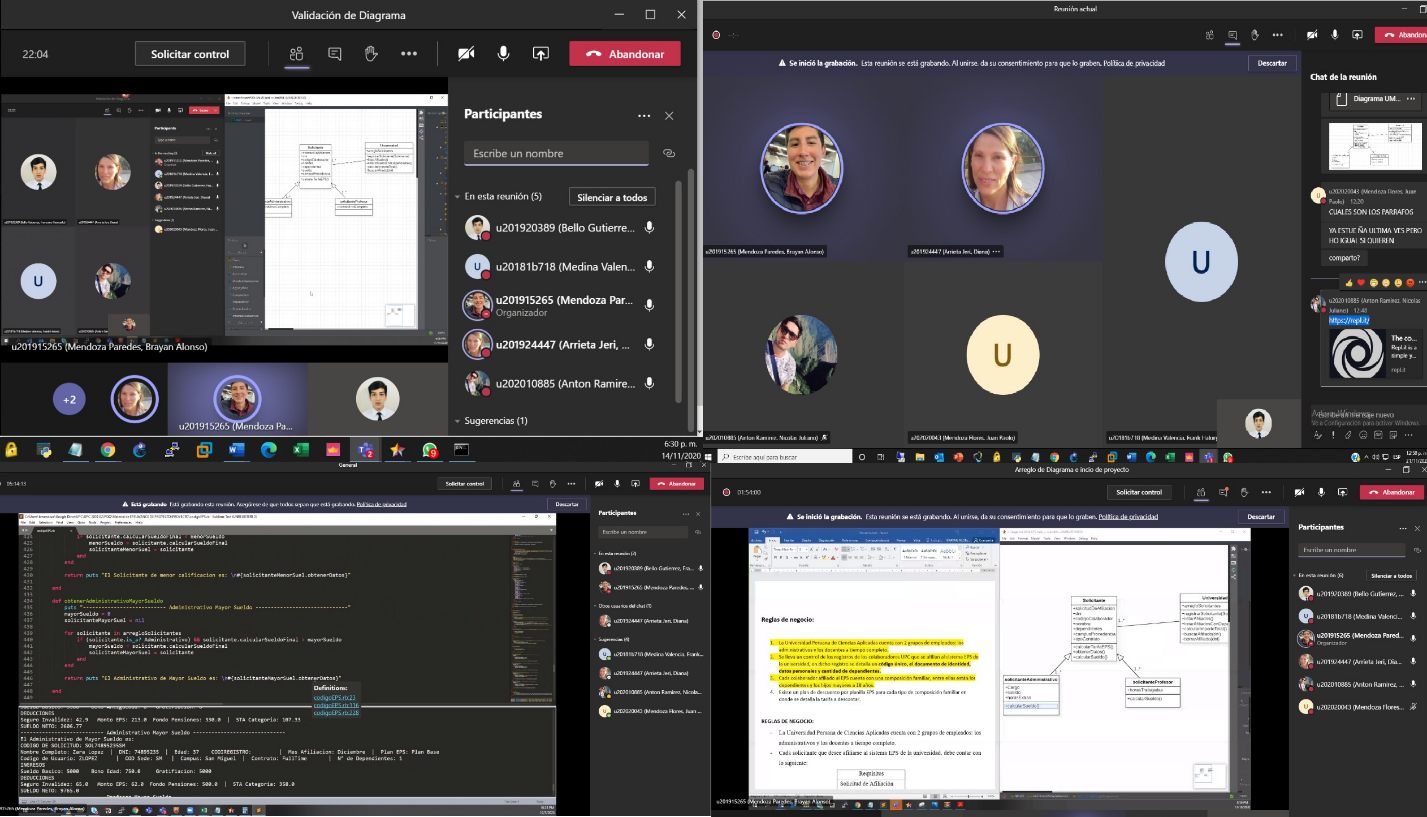
# DELEGANDO RESPONSABILIDAD



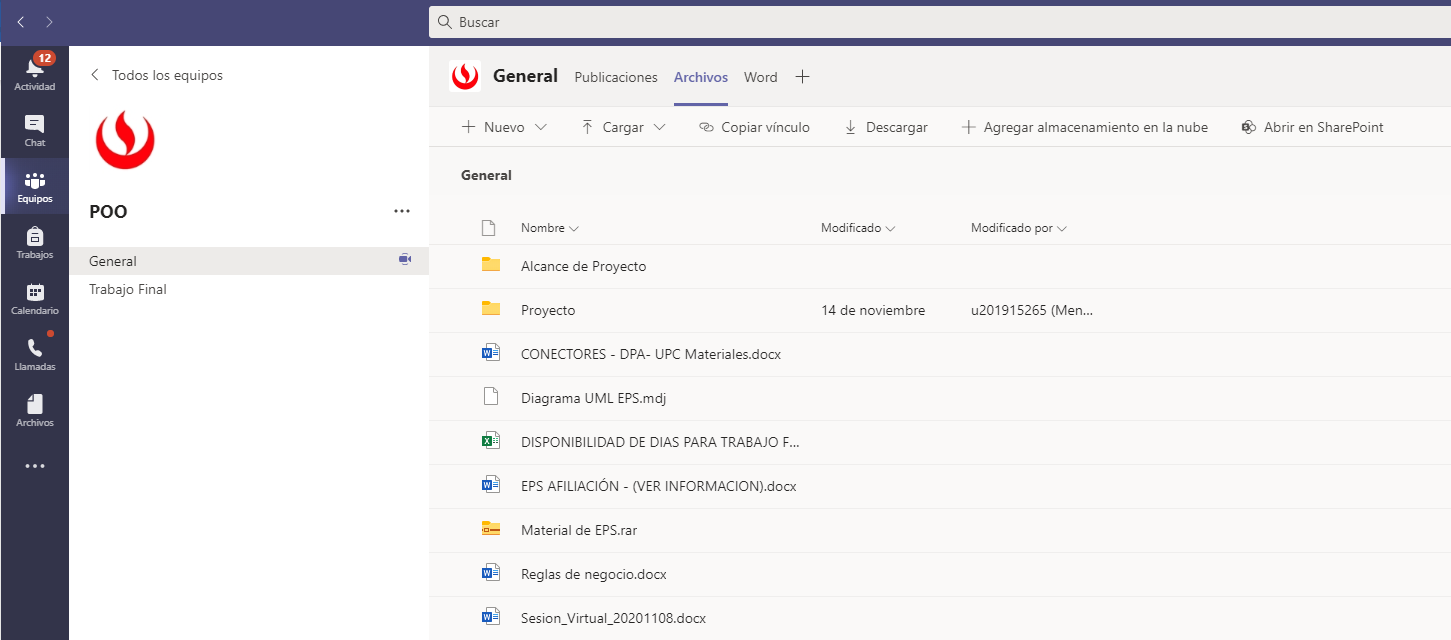
#REUNION [10]



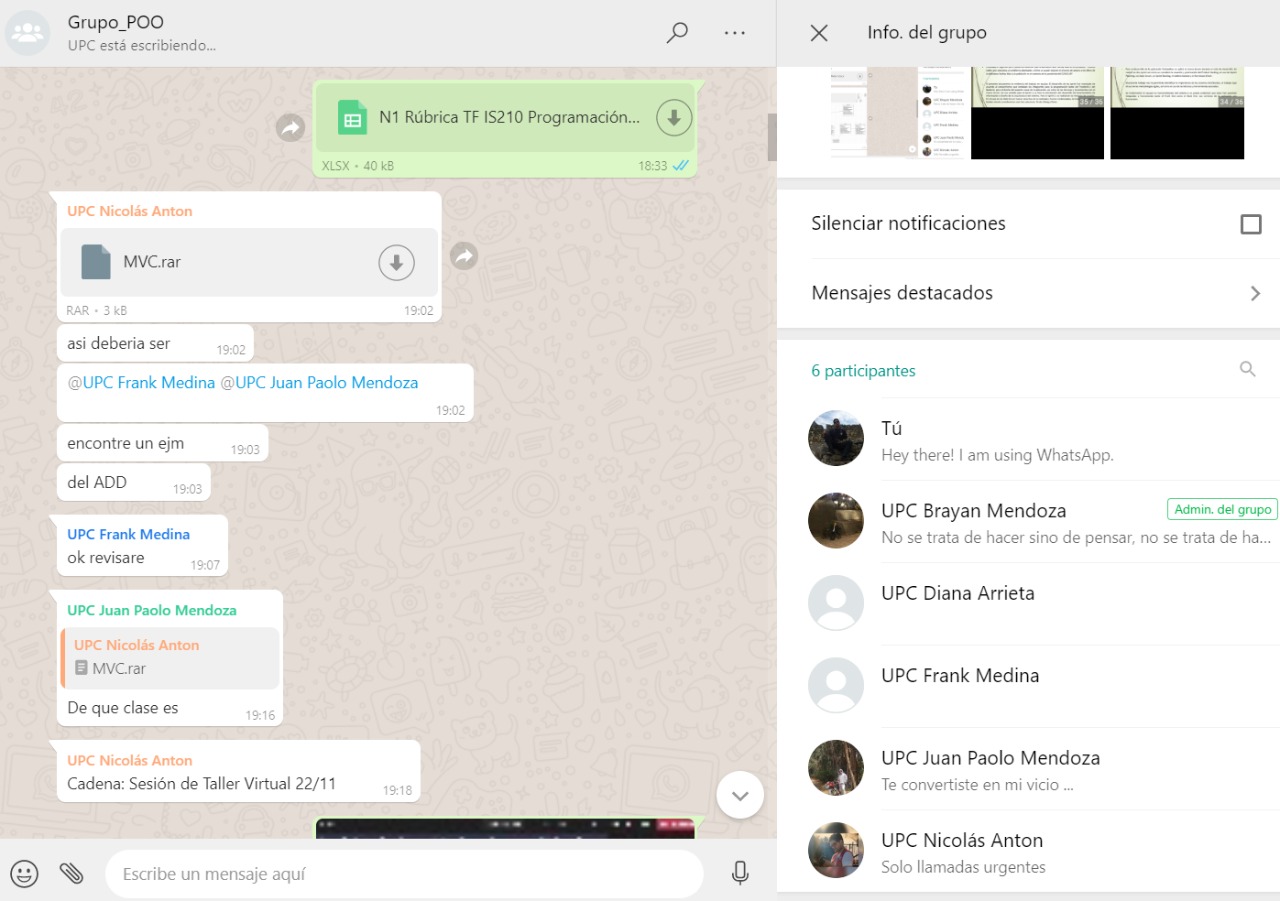
#REUNIONES



#ENTREGABLES EN REPOSITORIO TEAMS EDICION SIMULTANEA EN TIEMPO-REAL



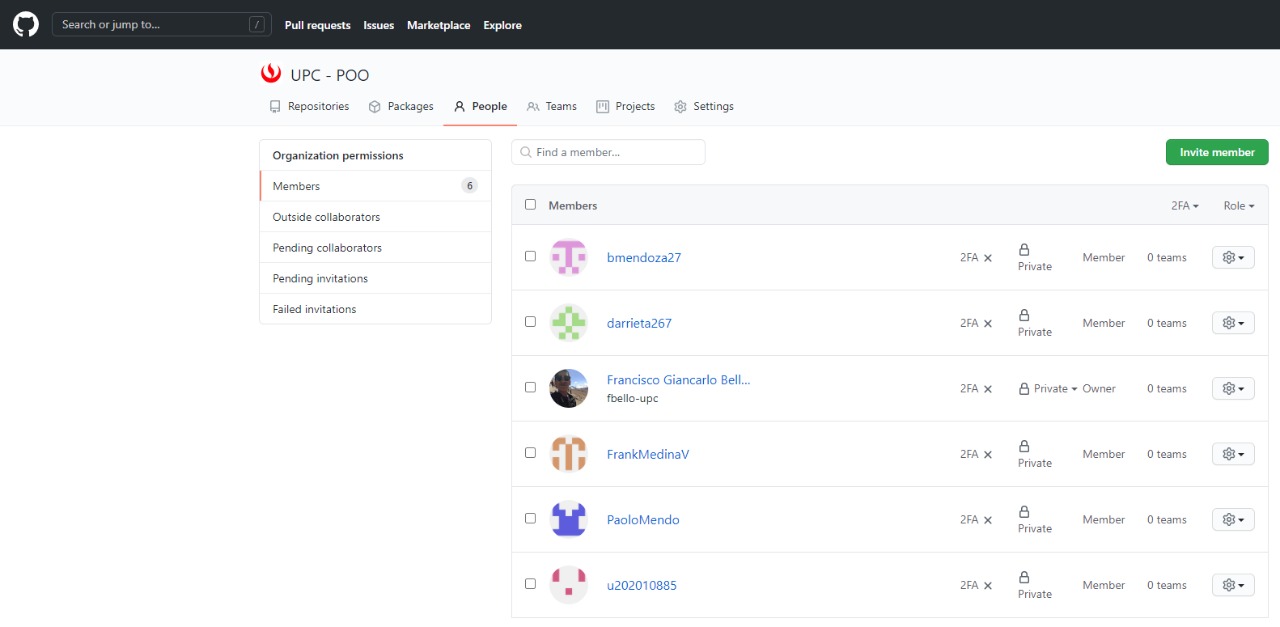
#MENSAJERIA INSTANTANEA vía WhatsApp

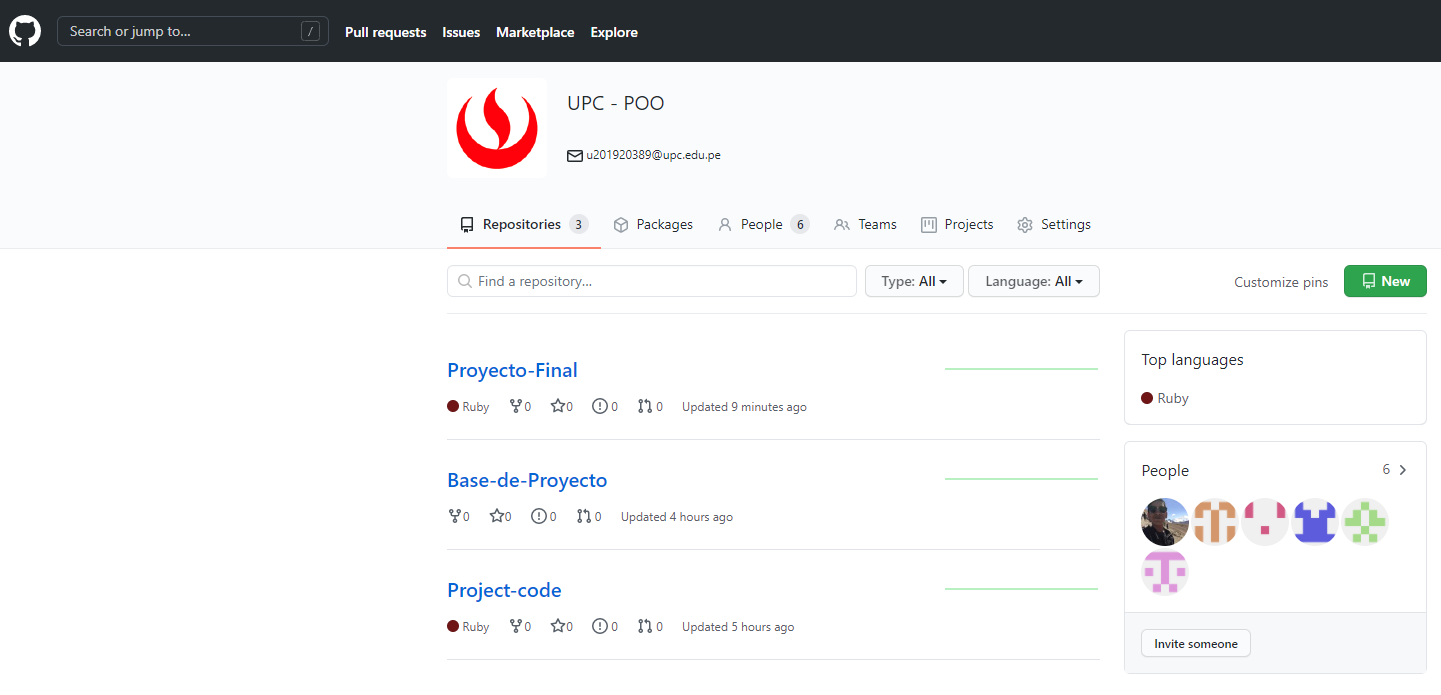




#REPOSITORIO Y CONTROL DE VERSIONES PARA DESARROLLO DE CODIGO en GITHUB

https://github.com/UPC-POO





## **CONCLUSIÓN**

Con este proyecto de gestión de afiliación de seguro automático hemos desarrollo de métodos de herencia para las clases hijas, y polimorfismo para los métodos heredados también con diferentes cálculos matemáticos.

Gracias a esto se logró la elaboración de un pequeño sistema para reportería. Se implementó la metodología PATRONES para que los espacios de memoria sean reutilizados por FACTORY y ahorrar líneas código para que esté se vea ama limpio u ordenador para futuros cambios o mejoras

Con el presente trabajo se logró aplicar los temas estudiados en el curso a un caso real, la programación orientada a objetos nos permitió desarrollar las clases y los métodos necesarios para el fin de nuestro trabajo: registro y evaluación y cálculo de los solicitantes al sistema EPS de la universidad.

## **RECOMENDACIONES**

En esta ocasión, como punto de recomendación se menciona lo siguiente:

* Reforzar el conocimiento en aplicar Patrones de Diseño.
* Reforzar el aprendizaje en la simulación del código en una aplicación o programa, ya que, durante las clases, no se ha tenido muy claro la ejecución del código en los distintos escenarios de los métodos. Por ejemplo, utilizando Factory, Singleton, Vista y Controlador.

## **BIBLIOGRÁFICA**

Repositorio de fuentes

<https://bitbucket.org/>

Documentación lenguaje ruby

<https://www.ruby-lang.org/es/>

Instalador de lenguaje ruby

<https://rubyinstaller.org/downloads/>

Plataforma de Directorio UPC

<https://conectate.upc.edu.pe/>

Herramienta de control de actividades

<https://trello.com/>

Plataforma GitHub

<https://github.com/UPC-POO>

Mensajeria Instantánea

https://web.whatsapp.com/

Plataforma de reunión

<https://www.microsoft.com/es-ww/microsoft-365/microsoft-teams/group-chat-software>

Información de 5ta Categoría

https://orientacion.sunat.gob.pe/index.php/personas-menu/impuesto-a-la-renta-personas-ultimo/rentas-de-quinta-categoria-personas/3071-02-calculo-del-impuesto