**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

**FACULTAD DE INGENIERÍA**

**DIVISIÓN DE ESTUDIOS PROFESIONALES PARA EJECUTIVOS**

**CARRERA DE INGENIERÍA DE SISTEMAS**

**PROYECTO PRESENTADO POR:**

**Ysabel Veronica Marcelo Suarez - U20151b867**

**César Renato Perales Reyna – U201500632**

**Victor Eduardo Torres Tejada - U20191b554**

**PROGRAMACION ORIENTADA A OBJETOS**

**PROFESOR:**

**Ing. Carlos Alberto Flores Orihuela**

**Lima, 03 de noviembre de 2019**

**ÍNDICE**

1. INTRODUCCIÓN 3

2. DIAGRAMA DE CLASES 4

3. URL DEL REPOSITORIO DE PROGRAMA FUENTE 5

4. RESULTADOS DE PRUEBAS UNITARIAS 6

5. CONCLUSIONES 9

6. RECOMENDACIONES 9

7. GLOSARIO 9

8. EVIDENCIAS DEL TRABAJO EN EQUIPO DE CADA PARTICIPANTE 10

9. BIBLIOGRAFIA 13

10. OBJETIVO DEL ESTUDIANTE (STUDENT OUTCOME) 13

# INTRODUCCIóN

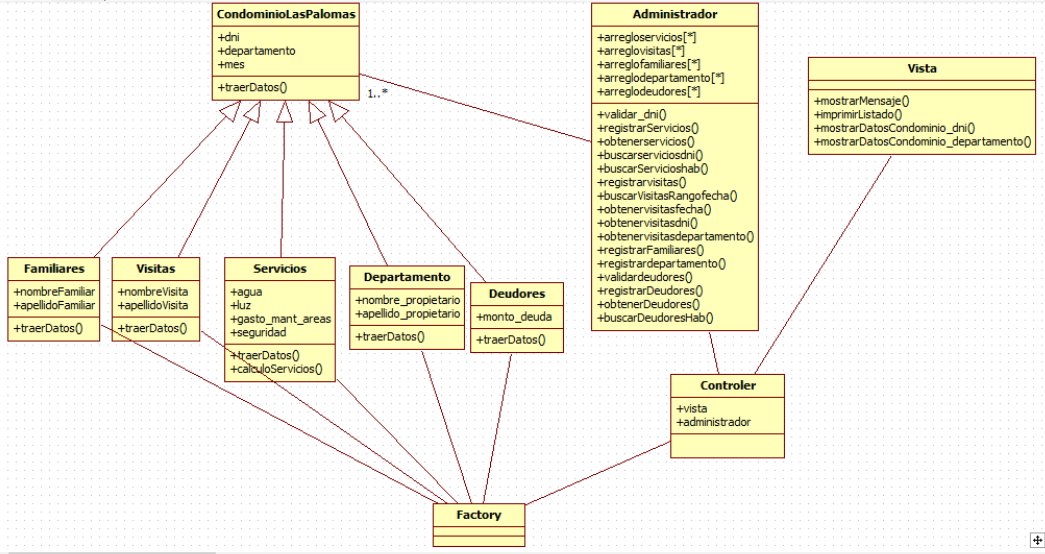
La constructora AM & Asociados es una empresa constructora de condominios y conjunto de departamentos. Debido al crecimiento en el sector construcción dado en los últimos años cada vez se construyen y se habitan más condominios. Debido a ello, la administración de estos condominios de una manera adecuada se ha convertido en un problema, para lo cual se requiere el empleo de una aplicación que cumpla con los siguientes requisitos funcionales.

* El sistema debe de calcular los pagos de mantenimiento (seguridad, luz, limpieza, áreas comunes, etc.).
* El sistema debe de contener a los familiares y visitantes autorizados por cada propietario de los departamentos.
* El sistema debe registrar los datos y fechas de visita de las personas autorizadas a cada departamento
* El sistema debe permitir realizar búsquedas por DNI de los visitantes
* El sistema deberá entregar el consolidado de deudores por los servicios de mantenimiento mensual

La empresa requiere el siguiente requisito de diseño:

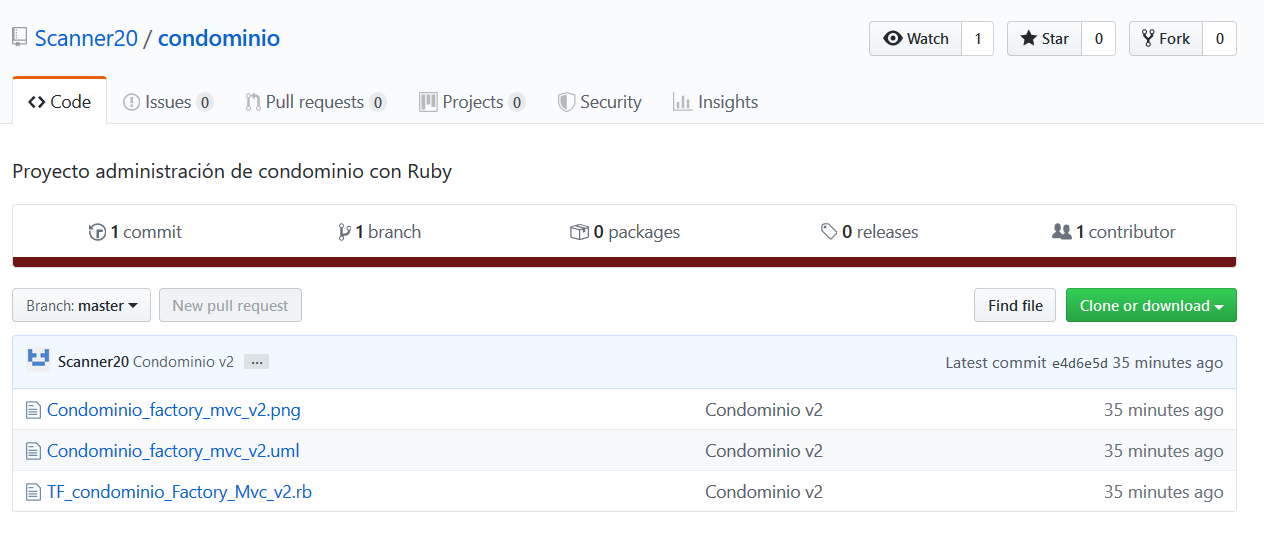
* Desarrollar las clases y sus relaciones.
* Aplique las pruebas unitarias a todos sus métodos o comportamientos de sus clases.
* Aplicar el patrón de diseño *Factory* para creación de objetos
* Aplicar el Patrón de diseño *Model View Controller*, MVC para todo el Programa.
* Controlar con excepciones la validación de todos los datos de entrada.

# DIAGRAMA DE CLASES



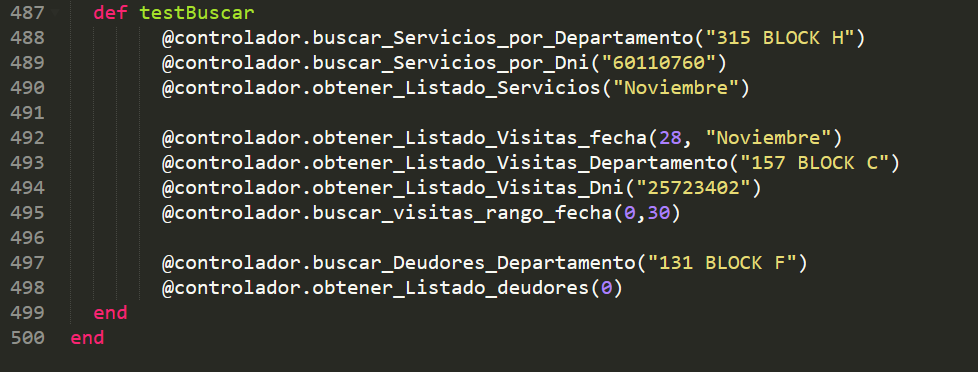
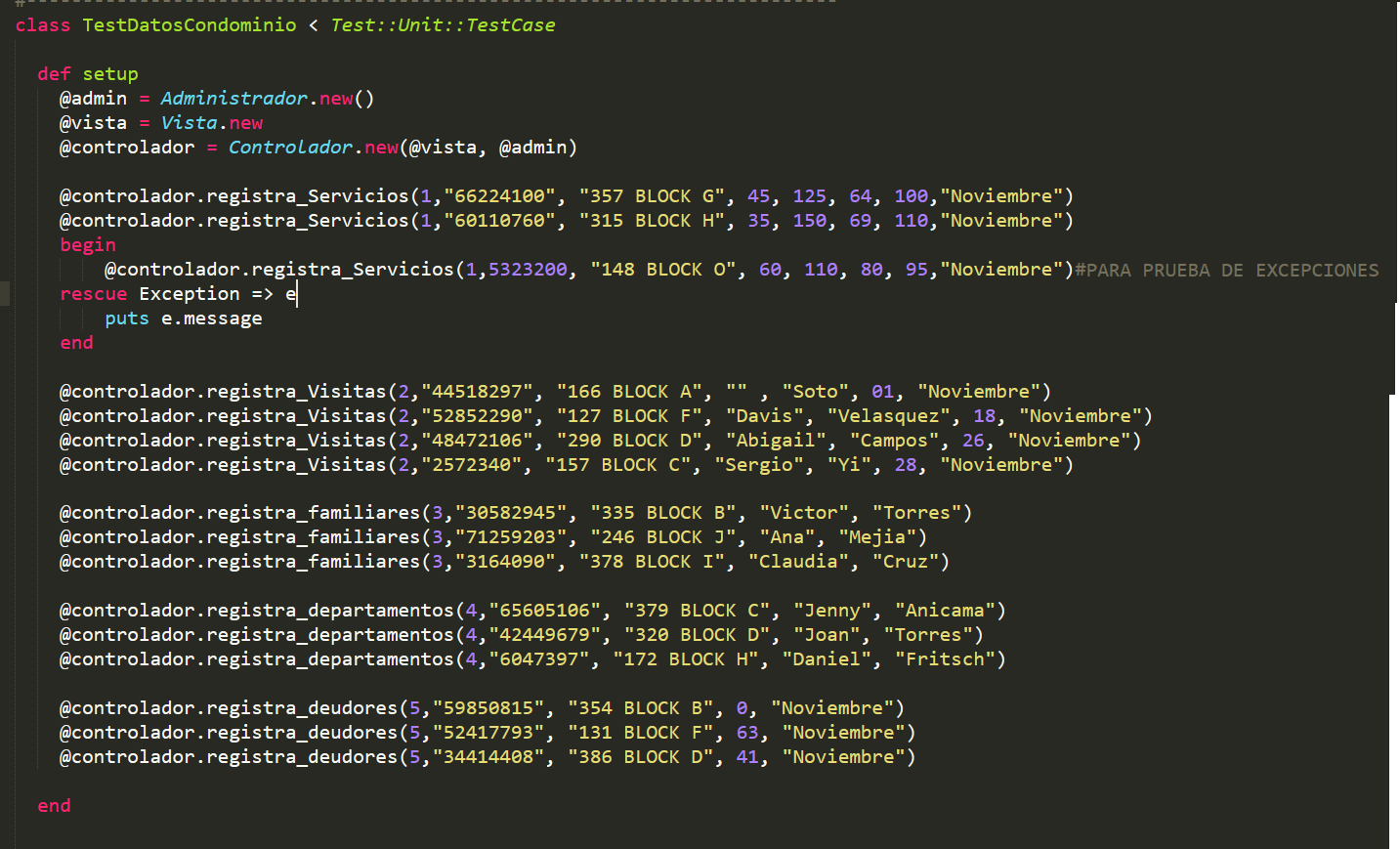
# URL del Repositorio de Programa Fuente

<https://github.com/Scanner20/condominio>

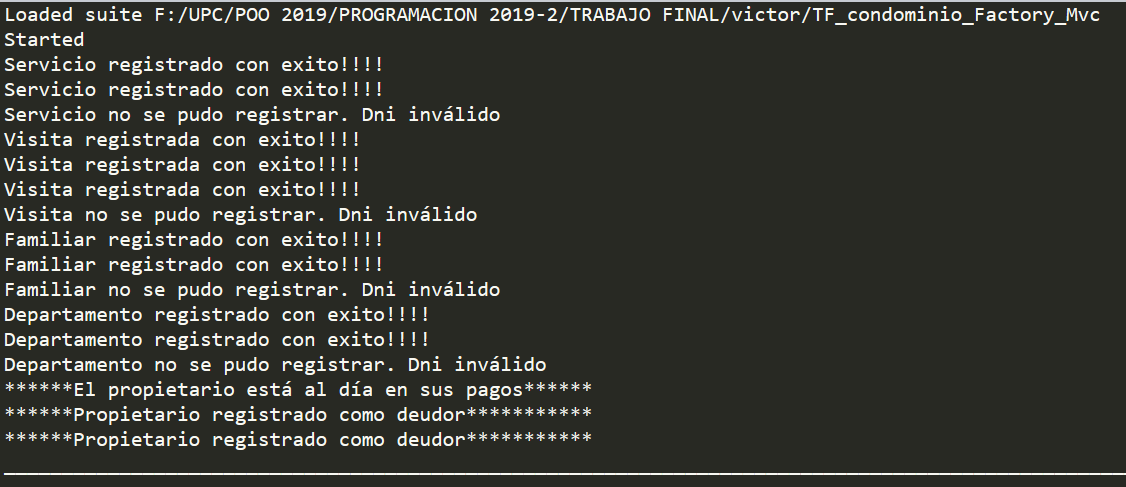


# Resultados de Pruebas Unitarias

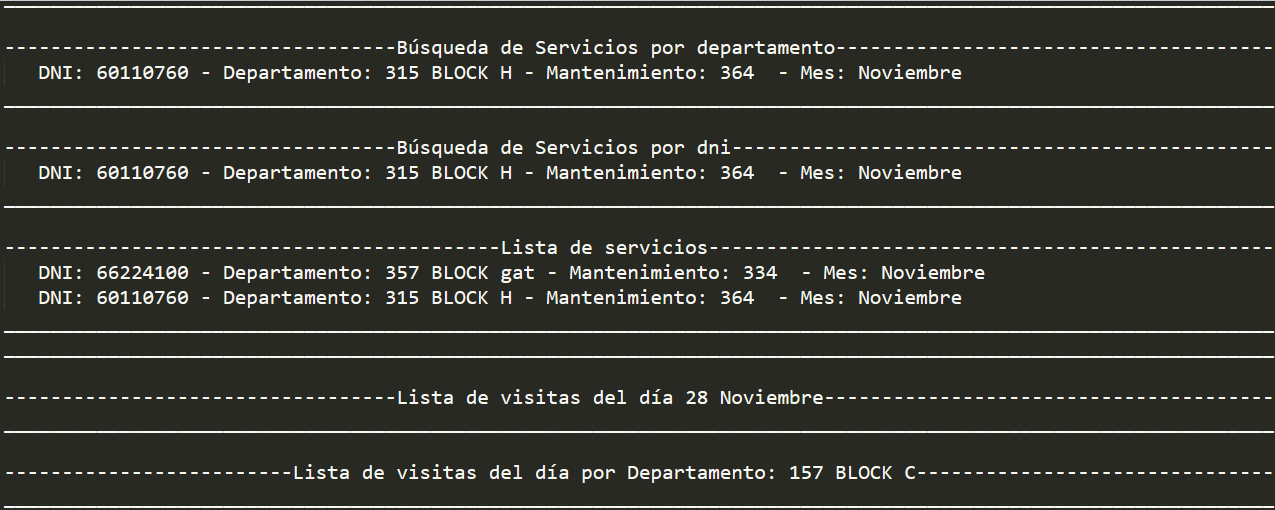
* Codigo de las pruebas unitarias

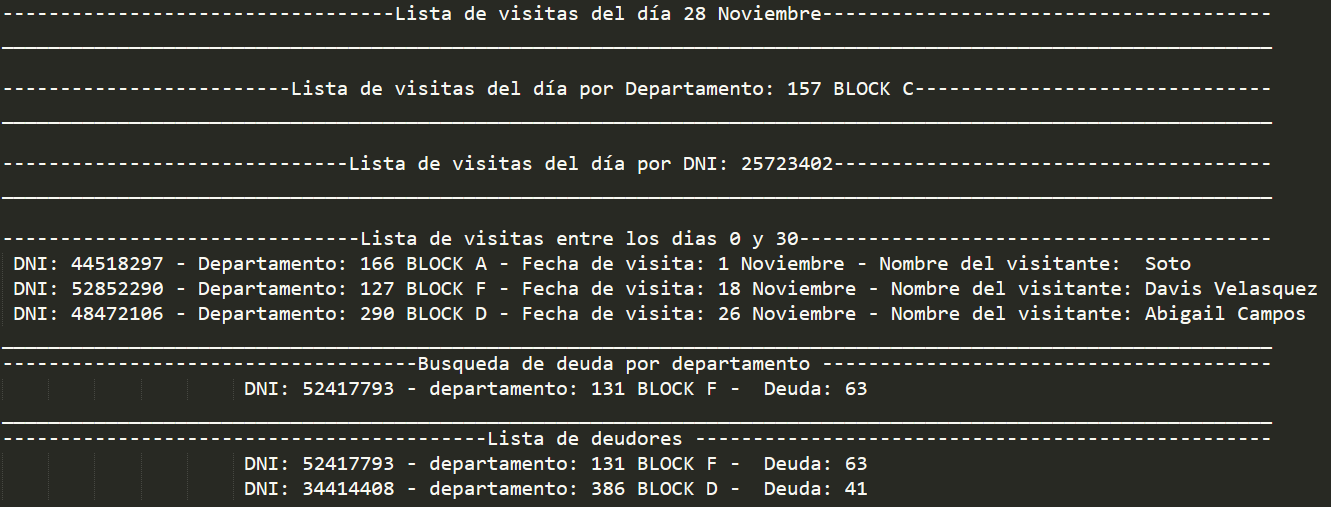


* Los registros se efectuaron de manera satisfactoria.

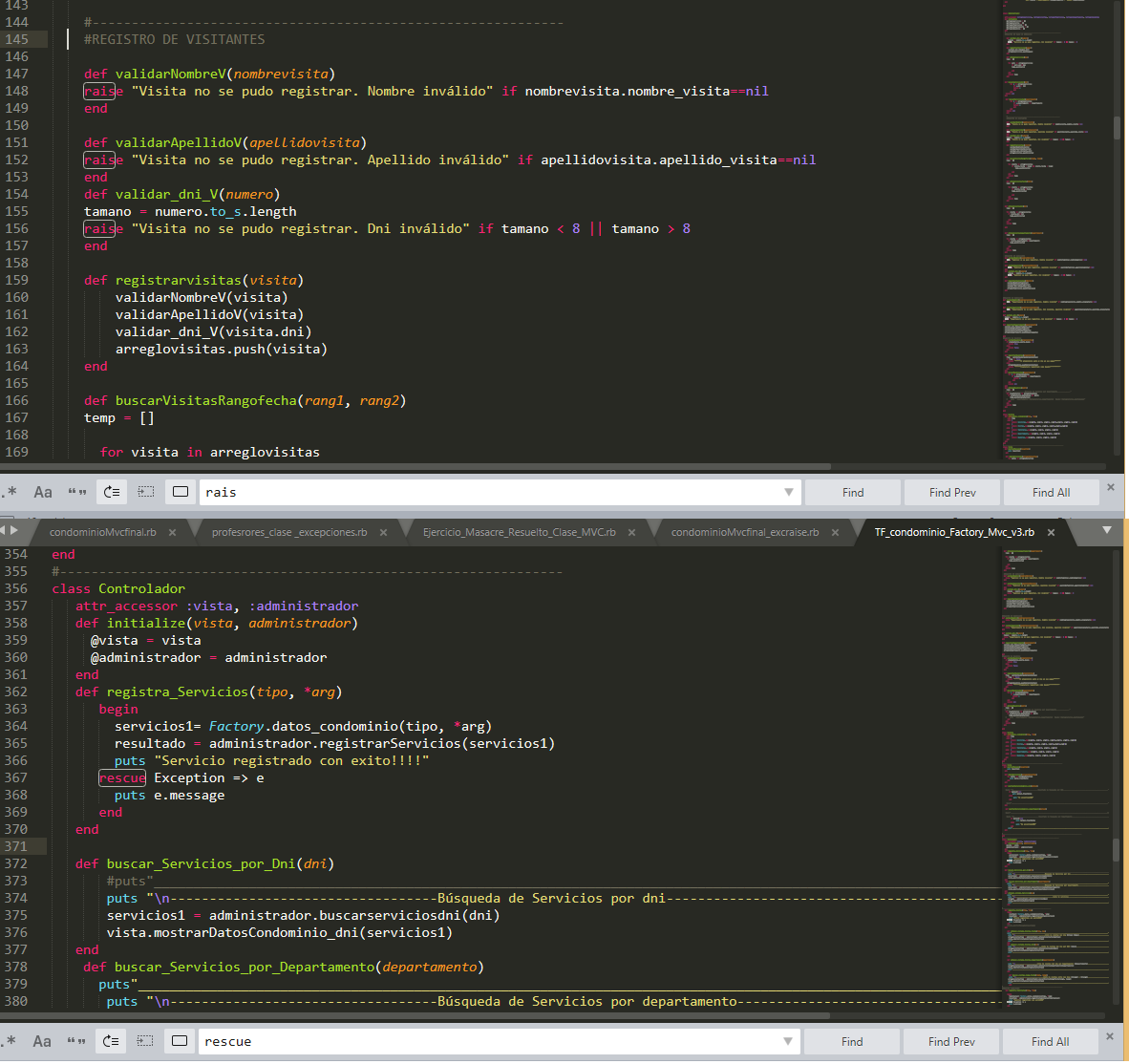


* Las búsquedas se realizaron de forma correcta

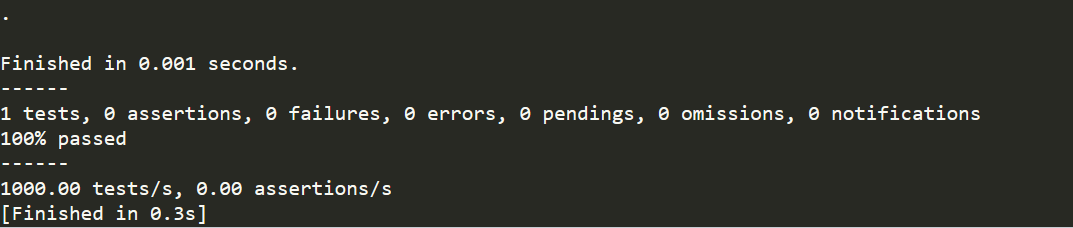




* Control de excepciones:



* Los resultados de las pruebas fueros satisfactorios



# Conclusiones

Dada la interdependencia que existe entre las áreas de una empresa, se sugiere identificar los procesos que se ejecutan de forma lenta o con poca o nula automatización y aislar los que dependen de transferencia de datos manuales para poder establecer prioridades entre los más críticos y vitales para establecer un cronograma de actualización y mejora de los procesos a través de los sistemas informáticos que los administran y creación de otros nuevos en caso no existan y se establezca su necesidad en el flujo de procesos principal de la empresa.

# Recomendaciones

Se puede interpretar cualquier regla de negocios en la administración de un condominio y desarrollar la automatización de cualquier tipo de proceso de provisión y/o cancelación de documentos conociendo el comportamiento de las reglas de negocio que utiliza una empresa de cualquier rubro y desarrollar aplicativos que ayuden a ingresar la información de manera más automatizada siendo el origen de información manual o formatos de trasferencia de datos como el XML. Se puede interpretar cualquier regla de negocios en la administración de un condominio y desarrollar la automatización de cualquier tipo de proceso de provisión y/o cancelación de documentos conociendo el comportamiento de las reglas de negocio que utiliza una empresa de cualquier rubro y desarrollar aplicativos que ayuden a ingresar la información de manera más automatizada siendo el origen de información manual o formatos de trasferencia de datos como el XML.

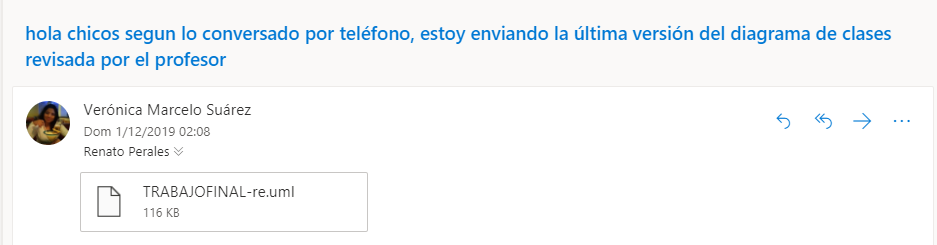
# Glosario

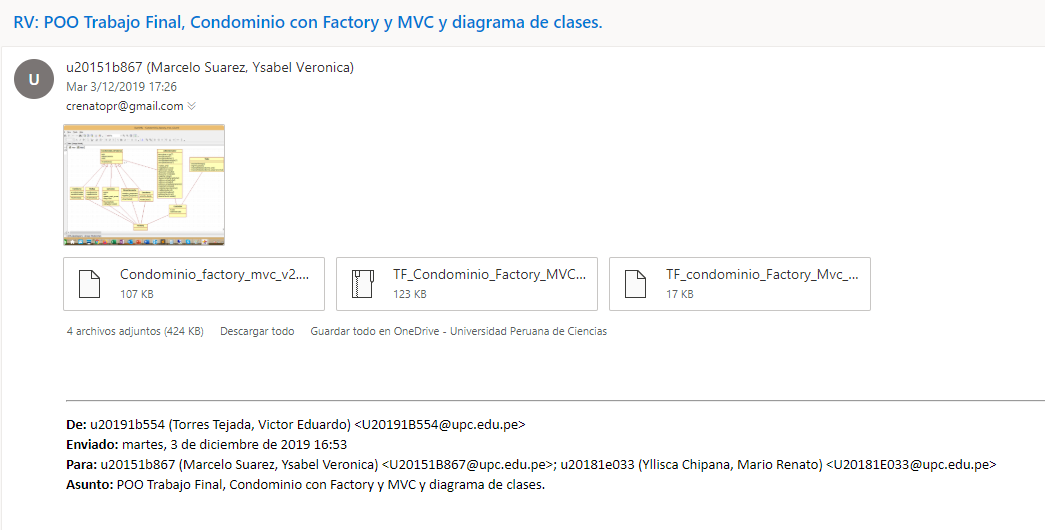
**MVC :** Es un patrón de arquitectura de software que se encarga de separar la lógica del negocio de la interfase del usuario.

**FACTORY:** el objetivo principal es que sea una clase “experta” .Es encargada de instanciar las clases concretas y no tenga que ser otra clase modelo la encargada de hacerlo.

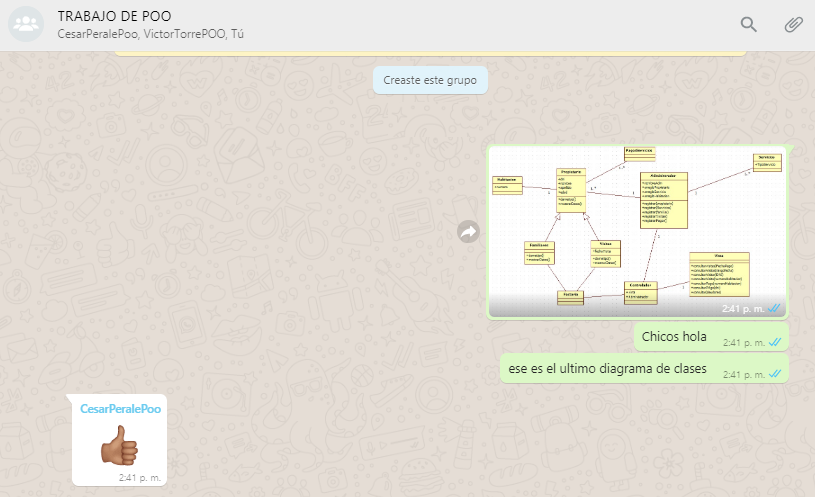
# Evidencias del Trabajo en Equipo

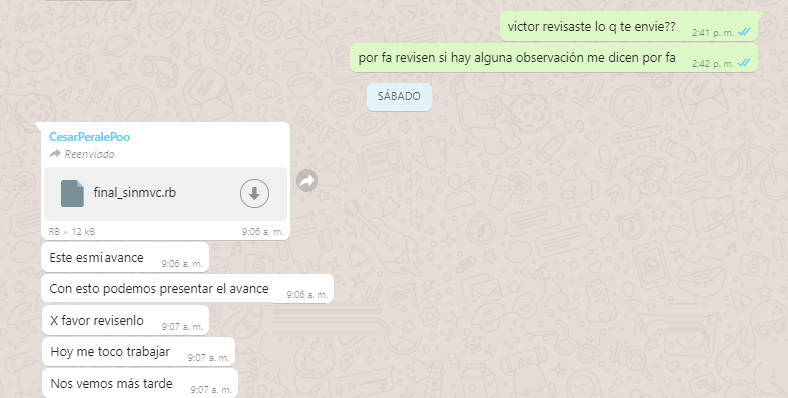
* Emails:

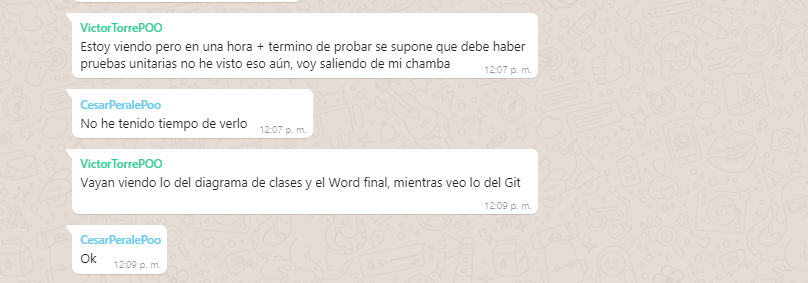


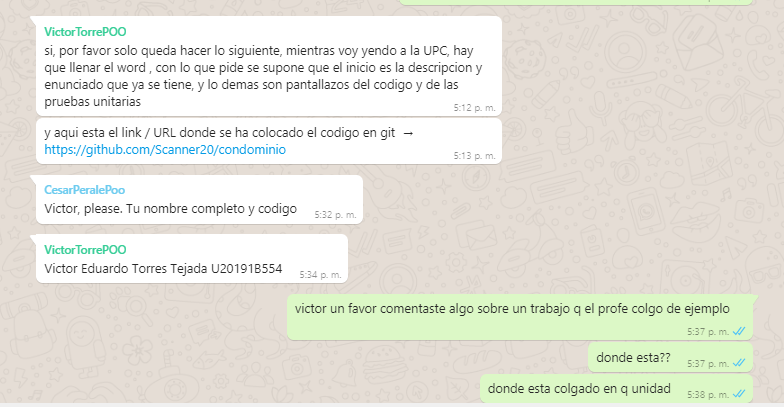


* Conversaciones de Whatsapp:









# BIBLIOGRAFIA

* The Ruby programming language / David Flanagan and Yukihiro Matsumoto
* Ruby Essentials - <https://www.techotopia.com/index.php/Ruby_Essentials>
* Learn to Program by Chris Pine - <https://pine.fm/LearnToProgram/?ShowTutorialCode=true>

# Objetivo del Estudiante (Student Outcome)

ABET – EAC - Student Outcome 5: La capacidad de funcionar efectivamente en un equipo cuyos miembros juntos proporcionan liderazgo, crean un entorno de colaboración e inclusivo, establecen objetivos, planifican tareas y cumplen objetivos.

El grupo logro funcionar en base a la responsabilidad de cada miembro, así como en la comunicación constante que tuvimos, lo cual a pesar de contar con poco tiempo para reunirnos, pudimos poder trabajar en equipo, dividirnos las tareas y cumplir con los objetivos trazados.