



# **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

## **FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

### **CARRERA DE INGENIERÍA EN SISTEMAS**

#### **“DESARROLLO DE LENGUAJE DE DOMINIO ESPECÍFICO PARA LA GENERACIÓN DE UNA INFRAESTRUCTURA CRUD- REST ESCRITA EN PYTHON Y FLASK CON CONEXIÓN A UNA BASE DE DATOS”**

**Trabajo de integración curricular**

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**AUTOR: GILBERTH IANROLANDO NÚÑEZ RIBADENEYRA**

**DIRECTOR: Dr. Omar Salvador Gómez Gómez**

Riobamba – Ecuador

2021

© 2021, **Gilberth Ianrolando Núñez Ribadeneyra**

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Gilberth Ianrolando Núñez Ribadeneyra, declaro que el presente trabajo de integración curricular es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de integración curricular; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 08 de junio de 2021

A handwritten signature in blue ink, appearing to read 'Gilberth Ianrolando Núñez Ribadeneyra', is written over a faint yellow rectangular stamp.

Gilberth Ianrolando Núñez Ribadeneyra  
020252150-6

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA EN SISTEMAS**

El Tribunal del Trabajo de Integración Curricular certifica que: El trabajo de integración curricular; tipo: Proyecto técnico, **DESARROLLO DE LENGUAJE DE DOMINIO ESPECÍFICO PARA LA GENERACIÓN DE UNA INFRAESTRUCTURA CRUD-REST ESCRITA EN PYTHON Y FLASK CON CONEXIÓN A UNA BASE DE DATOS**, realizado por el señor: **GILBERTH IANROLANDO NÚÑEZ RIBADENEYRA**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. Iván Menes Camejo <b>PRESIDENTE DEL TRIBUNAL</b>	 Firmado electrónicamente por: <b>IVAN MENES CAMEJO</b>	2021-06-08
Dr. Omar Salvador Gómez Gómez <b>DIRECTOR DE TRABAJO DE TITULACIÓN</b>	 Firmado electrónicamente por: <b>OMAR SALVADOR GOMEZ GOMEZ</b>	2021-06-08
Ing. Gloria Arcos Medina <b>MIEMBRO DEL TRIBUNAL</b>	 Firmado electrónicamente por: <b>GLORIA DE LOURDES ARCOS MEDINA</b>	2021-06-08

## **DEDICATORIA**

A mis padres Oscar Rolando Núñez Minaya y Melba Margoth Ribadeneyra Morales por su amor infinito, apoyo incondicional al ser en mi vida el pilar fundamental y la luz que guía mi horizonte, a mis hermanos Ronny Alejandro y Ramiro Marcelo quienes me han acompañado, apoyado y cuidado durante las diferentes etapas de mi vida. A todos mis familiares quienes con sus consejos y palabras de ánimo sembraron en mí el deseo de superación.

Gilberth

## **AGRADECIMIENTO**

A Dios por darme la salud y vida en esta época de pandemia, culminando con éxito mi carrera profesional.

A la Escuela Superior Politécnica de Chimborazo por abrirme sus puertas al conocimiento, a mis ilustres docentes de la Escuela de Ingeniería en Sistemas quienes con su saber, experticia y guía orientaron mi desarrollo en la programación en sistemas, durante mi vida estudiantil, un extensivo y fraterno agradecimiento al Dr. Omar Salvador Gómez Gómez, Director del presente trabajo de integración curricular quien con su sapiencia y conocimiento me orientó durante todo el desarrollo y culminación este trabajo de manera exitosa, a la Ing. Gloria Arcos quien con su conocimiento y guía termine mi trabajo de investigación.

Gilberth

## TABLA DE CONTENIDO

INDICE DE TABLAS.....	XII
ÍNDICE DE FIGURAS.....	XIV
ÍNDICE DE GRÁFICOS.....	XV
ÍNDICE DE ANEXOS .....	XVI
RESUMEN.....	XVII
SUMMARY .....	XVIII
INTRODUCCIÓN .....	1

### CAPITULO I

1	MARCO REFERENCIAL .....	2
1.1	Antecedentes .....	2
1.2	Formulación del problema .....	3
1.3	Sistematización del problema.....	3
1.4	Justificación .....	3
1.4.1	<i>Justificación teórica</i> .....	3
1.4.2	<i>Justificación aplicativa</i> .....	5
1.5	Objetivos .....	6
1.5.1	<i>Objetivo general</i> .....	6
1.5.2	<i>Objetivos específicos</i> .....	6

### CAPITULO II

2	MARCO TEÓRICO .....	7
2.1	Lenguaje de dominio específico (DSL) .....	7
2.1.1	<i>Definición</i> .....	7
2.1.2	<i>Ventajas y desventajas de los DSL</i> .....	8
2.1.3	<i>Análisis</i> .....	10
2.2	Python.....	10

2.2.1	<i>Definición</i> .....	10
2.2.2	<i>Características</i> .....	10
2.2.3	<i>Análisis</i> .....	11
2.3	<b>Flask</b> .....	11
2.3.1	<i>Definición</i> .....	11
2.3.2	<i>Comparativa con otro framework</i> .....	12
2.3.3	<i>Análisis</i> .....	12
2.4	<b>Pycharm</b> .....	13
2.4.1	<i>Definición</i> .....	13
2.4.2	<i>Características</i> .....	13
2.4.3	<i>Análisis</i> .....	15
2.5	<b>Entornos Virtuales</b> .....	15
2.5.1	<i>Definición</i> .....	15
2.5.2	<i>Problemas que soluciona</i> .....	15
2.5.3	<i>Análisis</i> .....	16
2.6	<b>PostgreSQL</b> .....	16
2.6.1	<i>Definición</i> .....	16
2.6.2	<i>Características</i> .....	17
2.6.3	<i>Análisis</i> .....	18
2.7	<b>MongoDB</b> .....	19
2.7.1	<i>Definición</i> .....	19
2.7.2	<i>Características</i> .....	19
2.7.3	<i>Análisis</i> .....	20
2.8	<b>Object-Relational mapping</b> .....	20
2.8.1	<i>Definición</i> .....	20
2.8.2	<i>Facilidades de un ORM.</i> .....	20
2.8.3	<i>Análisis</i> .....	21
2.9	<b>Psycopg2</b> .....	21



2.9.1	<i>Definición</i> .....	21
2.9.2	<i>Análisis</i> .....	22
2.10	SQLAlchemy.....	22
2.10.1	<i>Definición</i> .....	22
2.10.2	<i>Características</i> .....	22
2.10.3	<i>Análisis</i> .....	23
2.11	MongoAlchemy.....	23
2.11.1	<i>Definición</i> .....	23
2.11.2	<i>Análisis</i> .....	24
2.12	REST .....	24
2.12.1	<i>Definición</i> .....	24
2.12.2	<i>Restricciones de un sistema REST</i> .....	25
2.12.3	<i>Análisis</i> .....	26
2.13	Restful.....	26
2.13.1	<i>Definición</i> .....	26
2.13.2	<i>Diferencia entre los servicios Web</i> .....	27
2.13.3	<i>Análisis</i> .....	28
2.14	Serializadores.....	28
2.14.1	<i>Definición</i> .....	28
2.14.2	<i>Datos que pueden ser serializados con Python pickle</i> .....	28
2.14.3	<i>Análisis</i> .....	29
2.15	Marshmallow .....	29
2.15.1	<i>Definición</i> .....	29
2.15.2	<i>Análisis</i> .....	29
2.16	JSON.....	29
2.16.1	<i>Definición</i> .....	29
2.16.2	<i>Comparativa con XML</i> .....	30
2.16.3	<i>Análisis</i> .....	30

2.17	BSON.....	31
2.17.1	<i>Definición</i> .....	31
2.17.2	<i>Objetivos principales</i> .....	32
2.17.3	<i>Análisis</i> .....	32
2.18	Eclipse IDE.....	33
2.18.1	<i>Definición</i> .....	33
2.18.2	<i>Características</i> .....	33
2.18.3	<i>Análisis</i> .....	34
2.19	XText.....	35
2.19.1	<i>Definición</i> .....	35
2.19.2	<i>Características</i> .....	35
2.19.3	<i>Análisis</i> .....	36
2.20	XTend.....	36
2.20.1	<i>Definición</i> .....	36
2.20.2	<i>Ventajas y desventajas de Xtend</i> .....	37
2.20.3	<i>Análisis</i> .....	37
2.21	Productividad.....	38
2.21.1	<i>Definición</i> .....	38
2.21.2	<i>Análisis</i> .....	38

### CAPITULO III

3	MARCO METODOLÓGICO.....	39
3.1	Actividades de la metodología.....	39
3.1.1	<i>Tipo y métodos de investigación a utilizar</i> .....	39
3.2	Metodología para el desarrollo del DSL.....	40
3.3	Fase de planificación.....	41
3.3.1	<i>Personas y roles del proyecto</i> .....	42
3.3.2	<i>Product backlog</i> .....	42

3.3.3	<i>Sprint backlog</i> .....	44
3.3.4	<i>Planificación de la gestión de riesgos</i> .....	45
3.3.5	<i>Recursos necesarios</i> .....	46
3.4	<b>Fase de Desarrollo, generación de la aplicación modelo y definición de la gramática</b> .....	47
3.4.1	<i>Estándar de codificación</i> .....	47
3.4.2	<i>Desarrollo de los sprints</i> .....	48
3.4.3	<i>Estudio del dominio de aplicación del DSL</i> .....	52
3.4.4	<i>Implementación de la aplicación modelo (gold application)</i> .....	53
3.4.5	<i>Definición de la gramática del lenguaje</i> .....	58
3.4.6	<i>Generación del metamodelo</i> .....	59
3.4.7	<i>Implementación del generador de código</i> .....	62
3.5	<b>Ejemplificación de la propuesta de DSL</b> .....	62
3.5.1	<i>Gestión del proyecto</i> .....	64
3.5.2	<i>Gestión de los riesgos</i> .....	65

## CAPITULO IV

4	<b>MARCO DE RESULTADOS</b> .....	66
4.1	<b>Evaluar la mejora de la productividad al utilizar el DSL</b> .....	66
4.1.1	<i>Estudio 1: Conexión a una base de datos relacional</i> .....	66
4.1.2	<i>Estudio 2: Conexión a una base de datos no relacional</i> .....	71
4.1.3	<i>Análisis entre ambos estudios</i> .....	76

	<b>CONCLUSIONES</b> .....	77
--	---------------------------	----

	<b>RECOMENDACIONES</b> .....	78
--	------------------------------	----

## BIBLIOGRAFÍA

## ANEXOS

## INDICE DE TABLAS

<b>Tabla 1-2:</b> Ventajas de los DSL .....	8
<b>Tabla 2-2:</b> Ventajas de los DSL(Continuación de la tabla anterior) .....	9
<b>Tabla 3-2:</b> Desventajas de los DSL.....	9
<b>Tabla 4-2:</b> Características de Python .....	11
<b>Tabla 5-2:</b> Comparativa entre las características de Django y Flask .....	12
<b>Tabla 6-2:</b> Características de PyCharm.....	14
<b>Tabla 7-2:</b> Problemas que soluciona un entorno virtual.....	16
<b>Tabla 8-2:</b> Características de PostgreSQL .....	17
<b>Tabla 9-2:</b> Características de PostgreSQL (Continuación de la tabla anterior) .....	18
<b>Tabla 10-2:</b> Características de MongoDB.....	19
<b>Tabla 11-2:</b> Facilidades que proporciona un ORM.....	21
<b>Tabla 12-2:</b> Características de SQLAlchemy.....	23
<b>Tabla 13-2:</b> Restricciones que definen un sistema REST .....	25
<b>Tabla 14-2:</b> Restricciones que definen un sistema REST (Continuación de la tabla anterior) ..	26
<b>Tabla 15-2:</b> Diferencia entre los servicios Web RESTful y SOAP.....	27
<b>Tabla 16-2:</b> Datos que pueden ser serializados con Python pickle .....	28
<b>Tabla 17-2:</b> Comparativa con XML.....	30
<b>Tabla 18-2:</b> Objetivos principales de BSON.....	32
<b>Tabla 19-2:</b> Características de Python .....	34
<b>Tabla 20-2:</b> Características de Xtext.....	36
<b>Tabla 21-2:</b> Ventajas y desventajas de Xtend .....	37
<b>Tabla 1-3:</b> Roles y personas .....	42
<b>Tabla 2-3:</b> Planning poker.....	43
<b>Tabla 3-3:</b> Product backlog.....	43
<b>Tabla 4-3:</b> Sprint backlog.....	44
<b>Tabla 5-3:</b> Valores de impacto .....	45
<b>Tabla 6-3:</b> Tabla de riesgos.....	45
<b>Tabla 7-3:</b> Estrategias por riesgos.....	46
<b>Tabla 8-3:</b> Recurso hardware .....	46
<b>Tabla 9-3:</b> Recurso software .....	46
<b>Tabla 10-3:</b> Estándar de codificación.....	47
<b>Tabla 11-3:</b> Sprint 1 .....	48
<b>Tabla 12-3:</b> Sprint 2 .....	49
<b>Tabla 13-3:</b> Sprint 3 .....	49

<b>Tabla 14-3:</b> Sprint 4 .....	50
<b>Tabla 15-3:</b> Sprint 5 .....	50
<b>Tabla 16-3:</b> Sprint 6 .....	51
<b>Tabla 17-3:</b> Sprint 7 .....	51
<b>Tabla 18-3:</b> Sprint 8 .....	52
<b>Tabla 19-3:</b> Diccionario de datos de la tabla persona .....	56
<b>Tabla 1-4:</b> Contabilización de código fuente generado a partir de 59 LOC .....	68
<b>Tabla 2-4:</b> Contabilización de código fuente generado a partir de 59 LOC (Continuación de la tabla anterior).....	69
<b>Tabla 3-4:</b> Número de LOCs/hora .....	69
<b>Tabla 4-4:</b> Contabilización de código fuente generado a partir de 58 LOC .....	74
<b>Tabla 5-4:</b> Contabilización de código fuente generado a partir de 58 LOC (Continuación de la tabla anterior).....	75

## ÍNDICE DE FIGURAS

<b>Figura 2-1:</b> Diagrama UML de una arquitectura de un sistema basado en DSL .....	8
<b>Figura 2-2:</b> Representación de sistemas de una empresa que ofrecen silos de información....	25
<b>Figura 3-1:</b> Diagrama de Gantt .....	42
<b>Figura 3-2:</b> Estructura de un proyecto desarrollado en el lenguaje Python .....	53
<b>Figura 3-3:</b> Modelo lógico de la base de datos .....	55
<b>Figura 3-4:</b> Estructura de un proyecto modelo .....	56
<b>Figura 3-5:</b> Extracto del código de creación de clases.....	57
<b>Figura 3-6:</b> Extracto del código de la declaración de estructura de datos.....	57
<b>Figura 3-7:</b> Extracto del código de la gramática del DSL .....	59
<b>Figura 3-8:</b> Ecore metamodelo extracto del DSL .....	61
<b>Figura 3-9:</b> Extracto del código Xtend del generador de código implementado .....	62
<b>Figura 3-10:</b> Ejemplo de gramática del DSL.....	63
<b>Figura 3-11:</b> Extracto del código generado por el DSL.....	64
<b>Figura 4-1:</b> Ejemplo de gramática DSL con conexión a una BD relacional.....	67
<b>Figura 4-2:</b> Archivos generados a través de la gramática utilizada .....	68
<b>Figura 4 3:</b> Ejemplo de gramática DSL con conexión a una BD no-relacional.....	73
<b>Figura 4 4:</b> Archivos generados a través de la gramática utilizando conexión no relacional ..	74

## ÍNDICE DE GRÁFICOS

<b>Gráfico 1-3:</b> Gestión del proyecto.....	65
<b>Gráfico 1-4:</b> Histograma del tiempo de desarrollo del proyecto con BD relacional .....	71
<b>Gráfico 2-4:</b> Histograma del tiempo de desarrollo del proyecto con BD no-relacional.....	76

## ÍNDICE DE ANEXOS

<b>Anexo A:</b>	Base de datos
<b>Anexo B:</b>	Historias Técnicas
<b>Anexo C:</b>	Tareas de Ingeniería
<b>Anexo D:</b>	Pruebas de Aceptación
<b>Anexo E:</b>	Manual de usuario



## RESUMEN

Los lenguajes de dominio específico (DSL) son lenguajes de programación que fueron diseñados para resolver problemas en un dominio en particular con el objetivo de mejorar la productividad y la calidad en el proceso de desarrollo de sistemas, el presente trabajo de integración curricular tuvo como objetivo desarrollar un lenguaje de dominio específico basado en Python con un estilo de arquitectura REST. Para el desarrollo del DSL se tomó en cuenta la metodología propuesta por Gómez et al., para la construcción de este lenguaje. Las etapas que se desarrollaron en la metodología son: estudio del dominio de aplicación, implementación de la aplicación modelo, definición de la gramática del lenguaje, generación del metamodelo y la implementación del generador de código. Para la evaluación del DSL se desarrolló dos ejemplos que consistieron en el desarrollo de los módulos de una aplicación, que permitieron establecer la conexión a una base de datos relacional y una base de datos no relacional. Una vez evaluado el DSL se obtuvo datos de módulos de sistemas desarrollados con este lenguaje. Los hallazgos sugieren una tasa de automatización de rendimiento de 13,12 Líneas de Código (LOC) por cada LOC escrita en este DSL en sistemas con conexión a una base de datos relacional y de 10,53 LOC por cada LOC escrita en este DSL en sistemas con conexión a una base de datos no relacional

**Palabras clave:** <INGENIERIA DE SOFTWARE>, <LENGUAJE DE DOMINIO ESPECIFICO (DSL)>, <MAPEADOR DE RELACIÓN DE OBJETOS>, <SERIALIZACIÓN>.



Firmado electrónicamente por:  
**ELIZABETH  
FERNANDA AREVALO  
MEDINA**



1162-DBRAI-UPT-2021

## RESUMEN

Los lenguajes de dominio específico (DSL) son lenguajes de programación que fueron diseñados para resolver problemas en un dominio en particular con el objetivo de mejorar la productividad y la calidad en el proceso de desarrollo de sistemas, el presente trabajo de integración curricular tuvo como objetivo desarrollar un lenguaje de dominio específico basado en Python con un estilo de arquitectura REST. Para el desarrollo del DSL se tomó en cuenta la metodología propuesta por Gómez et al., para la construcción de este lenguaje. Las etapas que se desarrollaron en la metodología son: estudio del dominio de aplicación, implementación de la aplicación modelo, definición de la gramática del lenguaje, generación del metamodelo y la implementación del generador de código. Para la evaluación del DSL se desarrolló dos ejemplos que consistieron en el desarrollo de los módulos de una aplicación, que permitieron establecer la conexión a una base de datos relacional y una base de datos no relacional. Una vez evaluado el DSL se obtuvo datos de módulos de sistemas desarrollados con este lenguaje. Los hallazgos sugieren una tasa de automatización de rendimiento de 13,12 Líneas de Código (LOC) por cada LOC escrita en este DSL en sistemas con conexión a una base de datos relacional y de 10,53 LOC por cada LOC escrita en este DSL en sistemas con conexión a una base de datos no relacional

**Palabras clave:** <INGENIERIA DE SOFTWARE>, <LENGUAJE DE DOMINIO ESPECIFICO (DSL)>, <MAPEADOR DE RELACIÓN DE OBJETOS>, <SERIALIZACIÓN>.

## **ABSTRACT**

Domain-Specific Language (DSL) is a programming language that was designed to solve problems in a particular domain with the objective of boosting the productivity and the quality of the process of system development. The objective of the current curriculum integration work was to develop a domain-specific language based on Python with the software architectural style, REST. In order to develop the DSL, we took into consideration the methodology proposed by Gómez et al., for building this language. The phases that were developed in this methodology are: study of the application domain, implementation of the template application, definition of the grammar of the language, generation of metamodeling and implementation of code generation. We developed two samples for the DSL evaluation which consisted in the development of modules of an application that allowed establishing the connection to a relational data base and to a non-relational database. Once the DSL was evaluated, we obtained data of modules of systems developed with this language. The results suggest a performance automation rate of 13,12 Lines of Code (LOC) for each LOC written in this DSL in systems with connection to a relational data base and 10,53 LOC for each LOC written in this DSL in systems with connection to a non-relational data base.

**Keywords:** <SOFTWARE ENGINEERING>, < DOMAIN-SPECIFIC LANGUAGE (DSL)>, <OBJECT-RELATIONAL MAPPING>, <MARSHALLING>.

## INTRODUCCIÓN

Los lenguajes de dominio específico (DSL) han sido parte de la computación desde hace mucho tiempo, son muy útiles en agilizar los procesos de desarrollo y facilitar la comunicación, a pesar de esto, pocas veces se los toma en cuenta al momento del desarrollo de software y como consecuencia de esto actualmente no se dispone de una *API Rest* en la cual se pueda apoyar para el desarrollo de proyectos software en el lenguaje *Python*.

Este trabajo consistió en el desarrollo de lenguaje de dominio específico (DSL) para generar de manera automática recursos *REST* a partir de operaciones *CRUD* y de esta manera los desarrolladores dispongan de una herramienta con estas características para apoyarse en el proceso de desarrollo de software.

Para el desarrollo de este trabajo fue necesario el uso de dos metodologías, la metodología propuesta por Gómez et al. fue preciso para el desarrollo del DSL ya que en esta se detalla las etapas que deben ser efectuadas al momento de realizar un proyecto de este tipo y para la gestión del proceso de desarrollo se utilizó la metodología SCRUM.

En el presente trabajo de integración curricular se describen los siguientes capítulos:

**Capítulo I**, es el Diagnóstico Del Problema, el que contiene antecedentes, formulación y sistematización del problema, justificación teórica, justificación aplicativa, además de los objetivos generales y específicos.

**Capítulo II**, abarca los Fundamentos Teóricos, en el que se describe los temas recopilados para el desarrollo e implementación del proyecto técnico.

**Capítulo III**, es el Marco Metodológico, que abarca las fases de la metodología propuesta por Gómez et al. y la metodología ágil SCRUM para la gestión del proceso desarrollo del software.

**Capítulo IV**, es de Resultados, que contiene los resultados obtenidos luego de evaluar la productividad en el desarrollo de software utilizando el DSL propuesto.

# CAPÍTULO I

## 1 Definición del Problema

En este capítulo se presenta los antecedentes, la formulación del problema, sistematización del problema, justificación teórica, justificación aplicativa, así como también el objetivo general y los objetivos específicos.

### 1.1 Antecedentes

Después del surgimiento de los servicios REST en el año 2000 se cambió por completo la ingeniería de software, Roy Fielding lo presentó en la Universidad de California por primera vez, brindando un nuevo enfoque de desarrollo de proyectos y servicios web. Una definición sencilla es la que nos provee Javier Ceballos en su libro “JAVA. Interfaces gráficas y aplicaciones para Internet” en el cual se menciona que “...describe a cualquier interfaz que transmite datos específicos de un dominio sobre HTTP sin una capa adicional...” (Sierra, 2015, p. 683) no hay que confundir *REST* con un estándar ya que este es un estilo de arquitectura basado en estándares.

Es por ello por lo que actualmente se disponen de *Apis REST* para la creación de servicios y miles de empresas generan negocio gracias a estas, sin ellas todo el crecimiento de cualquier negocio sería imposible.

Con la actual popularidad alcanzada por el lenguaje *Python* varias empresas optaron por migrar sus aplicaciones a *Python*, empresas desarrolladoras de software decidieron incluir este lenguaje para el desarrollo de futuros proyectos. Para el desarrollo de aplicaciones web se cuenta varios *frameworks* de los cuales se tomaron en cuenta *Flask* debido a su flexibilidad.

En la actualidad no se dispone de una *API Rest* en la cual se pueda apoyar para el desarrollo en este lenguaje pero para la ejecución del presente trabajo se tomaron en cuenta desarrollos de DSL previos como “Diseño de un prototipo de DSL basado en ingeniería de modelos para el control a distancia de un dron terrestre” (Valencia, 2017), como también en el área de educación el uso de un MDE apoyado por un DSL reduce el tiempo y esfuerzo en la creación y despliegue de módulos que se presenta en el trabajo “Desarrollo de un lenguaje de dominio específico para sistemas de gestión de aprendizaje y su herramienta de implementación “KiwiDSM” mediante ingeniería dirigida por modelos” (Montenegro et al., 2010), se ha desarrollado un lenguaje para definir las métricas deseadas en un metamodelo el cual nos permitió contar cierto tipo de objetos hasta

condicionar las cuentas mencionadas “Un lenguaje de dominio específico para la definición de métricas para lenguajes de modelado” (Cobos, 2017).

El problema es que toma demasiado tiempo generar los servicios REST en forma manual, lo cual es causado por la falta de herramientas que apoyen el proceso, la falta de capacitación al personal de desarrollo, el no utilizar estándares en el código como efecto de ello se presentan pérdidas económicas y disminución en la productividad, desconocimiento del desarrollo de un lenguaje de dominio específico, código ilegible y difícil de entender el cual en un futuro complica el mantenimiento del código.

Por esta razón se realizó el presente trabajo de integración curricular que consiste en el desarrollo de lenguaje de dominio específico para generar de manera automática recursos *REST* a partir de operaciones *CRUD* para ello se trabajó con el lenguaje de programación *Python*, se utilizó el framework *Flask*, el *IDE* de desarrollo será Eclipse, para la conexión con la base de datos se trabajó tanto con *Postgres* como base de datos relacional y *MongoDB* como no relacional.

## **1.2 Formulación del problema**

¿El desarrollar un lenguaje de dominio específico (DSL) para generar de manera automática recursos REST en el lenguaje *Python* con el *framework Flask* a partir de operaciones *CRUD* tanto en una base de datos relacional como una no relacional aumenta la productividad en el desarrollo de software?

## **1.3 Sistematización del problema**

- ¿Qué herramientas para construir lenguajes de dominio específicos existen actualmente?
- ¿Cómo abstraer el proceso llevado para la construcción de un servicio *REST*?
- ¿Qué tipo de entrada interpretará el DSL?
- ¿Cómo mejorar la productividad el lenguaje de dominio específico propuesto?

## **1.4 Justificación**

### ***1.4.1 Justificación teórica***

Las tecnologías de la información y de la comunicación (TIC) tienen el objetivo de simplificar el esfuerzo necesario para realizar una o varias tareas, de modo que esta área ha innovado constantemente al ser de gran apoyo para el ser humano en diversas actividades tanto en cosas básicas a modo de pequeños cálculos, así como también acortando grandes distancias con el

Internet y de esta manera poner a disposición toda la información que existente en la actualidad. Es así como hablan las autoras Virna Gil , Xantipa Rosas y Marisol Olmeda en su libro “Fundamentos de las tecnologías de Información: viviendo en una sociedad tecnológica” nos indican que “Las tecnologías de información extienden las posibilidades de comunicación del ser humano, ya que, por sus características, permiten intercambiar y distribuir información en forma instantánea, entre personas y organizaciones, utilizando redes de comunicación” (Castro et al., 2014), es por ello que actualmente disponemos de tanta información y facilidad para acortar distancia con el uso de las redes de comunicación.

En cuanto al desarrollo de software se refiere Pohjonen y Kelly en su publicación denominada “Domain-specific modeling” al modelado específico de dominio y su importancia al eliminar varios problemas que aún persisten en esta área, ellos expresan:

*Tradicionalmente, el desarrollo de software ha sido una serie de mapeos desde la idea de dominio, hasta los modelos de diseño y el código fuente. Estos mapeos tienden a ser lentos y conducen a errores y duplicación de esfuerzos en la resolución de problemas, el diseño y la codificación. El modelado específico del dominio (DSM) aborda estos problemas eliminando los mapeos intensivos en recursos y propensos a errores, con el objetivo de resolver el problema una sola vez al mismo nivel de abstracción con el dominio mismo (Pohjonen & Kelly, 2002, p. 1)*

Los lenguajes de dominio específico (DSL) han sido parte de la computación desde hace mucho tiempo y han sido muy útiles en agilizar los procesos de desarrollo y facilitar la comunicación, a pesar de esto, pocas veces se los toma en cuenta al momento del desarrollo de software es por ello que Martin Fowler en su libro “Domain Specific Languages” expresa que la mayoría de la gente aprende acerca de los DSL de otra persona y que muchas veces solo aprenden un conjunto de técnicas muy limitado.

Las características que resalta Fowler sobre los DSL son:

*Mejorar la productividad de los desarrolladores y mejorar la comunicación con los expertos en dominios. Un DSL bien elegido puede hacer más fácil la comprensión de un bloque de código complicado, mejorando así la productividad de los que trabajan con ella. También puede facilitar la comunicación con expertos en el dominio, proporcionando un texto común que actúa tanto como un programa ejecutable y una descripción que los expertos en dominios pueden leer para entender cómo sus ideas están representadas en un sistema. Esta comunicación con los expertos del dominio es un beneficio más difícil de lograr, pero la ganancia resultante es mucho más amplia porque ayuda a desatascar uno de los peores cuellos de botella en el desarrollo de software, la comunicación entre los programadores y sus clientes (Fowler, 2010).*

Es gracias a estas características que el desarrollo de un DSL mejora significativamente la productividad de un equipo de desarrollo eliminando los cuellos de botella como menciona Fowler.

Para el desarrollo de aplicaciones web se cuenta varios *frameworks* de los cuales se tomó en cuenta *Flask* debido a la flexibilidad, simplicidad y gran control que nos proporciona y es por la primera característica que se optó por dicho *framework* ya que nos permitió crear nuestros servicios web *REST*.

En virtud de lo anterior, se consideró relevante el desarrollo del presente trabajo de integración curricular el cual permitió reducir los tiempos de desarrollo y podrá contribuir en los futuros sistemas que opten por la utilización de esta herramienta.

#### **1.4.2 Justificación aplicativa**

El tiempo que toma el desarrollo de los servicios web podría ser usado para otras funcionalidades, esto resta productividad al equipo de desarrollo. Actualmente existen *APIs REST* que resuelven esta problemática en otros lenguajes, pero al hablar del lenguaje Python nos encontramos con la limitante de no existir herramientas que generen estos servicios por lo que para solucionar este inconveniente se desarrolla el DSL que permite la automatización de los recursos CRUD-REST en el lenguaje Python

A continuación, se detalla las etapas que se cumplió para el desarrollo del DSL propuestas por Gómez et al.

- Estudio del dominio de aplicación del DSL
- Implementación de la aplicación modelo (gold application)
- Definición de la gramática del lenguaje
- Generación del metamodelo
- Implementación del generador de código
- Evaluar la mejora de la productividad al utilizar el DSL propuesto

El lenguaje de dominio específico contribuirá a la Escuela Superior Politécnica de Chimborazo con la automatización de la generación de código reduciendo así, el tiempo necesario para el desarrollo del BackEnd de los sistemas que opten por utilizar el DSL propuesto, aportando de esta forma con el Plan Nacional de Desarrollo según el objetivo cinco que consiste en impulsar la productividad y competitividad para el crecimiento económico sostenible de manera



redistributiva y solidaria (Secretaría Nacional de Planificación y Desarrollo - Senplades, 2017), además se vincula a las líneas de investigación de la ESPOCH 2018-2022 con respecto a las Tecnologías de la Información y Comunicación, que está ligada con la línea de investigación de Ingeniería de Software de la escuela de Ingeniería en Sistemas de la Escuela Superior Politécnica de Chimborazo.

## **1.5 Objetivos**

### ***1.5.1 Objetivo general***

Desarrollar un lenguaje de dominio específico (DSL) para generar de manera automática recursos REST en el lenguaje Python con el framework Flask a partir de operaciones CRUD tanto en una base de datos relacional como una no relacional para aumentar la productividad.

### ***1.5.2 Objetivos específicos***

- Analizar herramientas para construir lenguajes específicos de dominio.
- Desarrollar la aplicación modelo para abstraer el proceso llevado a cabo en la construcción de un servicio REST.
- Desarrollar la gramática para que el DSL la interprete.
- Evaluar la productividad en el desarrollo de software utilizando el DSL propuesto.

## CAPÍTULO II

### 2 MARCO TEÓRICO

En el presente capítulo se presenta las bases teóricas, así como también la definición de los conceptos.

#### 2.1 Lenguaje de dominio específico (DSL)

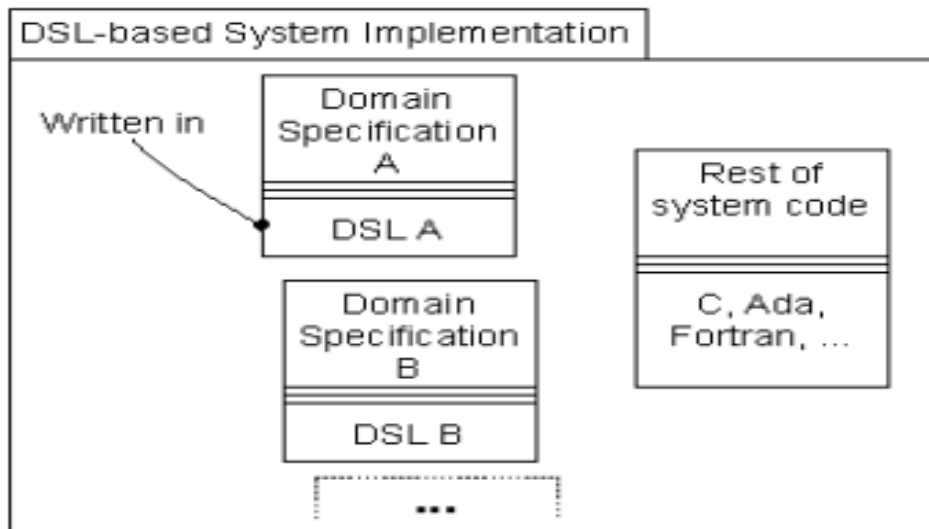
##### 2.1.1 Definición

Varias son las razones que existen actualmente para el uso de los DSLs y dentro de las principales se menciona el hecho de que mejora la productividad de los desarrolladores, así como también mejora la comunicación.

*Los DSL intercambian generalidad por expresividad en un dominio limitado. Ofrecen notaciones y construcciones orientadas a un dominio particular, presentando ganancias sustanciales en términos de expresividad y facilidad de uso en comparación con los lenguajes de propósito general (GPL) para el dominio en cuestión. Estas ganancias se corresponden en mejoras de la productividad y reducción de los costes de mantenimiento. (Montenegro et al., 2010, p. 71)*

El uso de DSLs nos permite facilitar el entendimiento de un bloque de código, al mismo tiempo que nos permite escribir y modificar código rápidamente.

*Un lenguaje específico del dominio es un lenguaje de programación que está diseñado para utilizarse en dominios o problemas específicos, a diferencia de los lenguajes de programación de propósito general. Tienen un mayor nivel de abstracción que los lenguajes base y expresan los conceptos de dominio específico en un nivel de representación más alto. Representan los elementos de plataformas de implementación. (Mora et al., 2006, p. 30)*



**Figura 2-1:** Diagrama UML de una arquitectura de un sistema basado en DSL

Fuente: (Mora et al., 2006, p. 31)

### 2.1.2 Ventajas y desventajas de los DSL

Martin Fowler en su libro “Domain-Specific Languages” manifiesta que existen varias razones por las cuales se debería proponer implementar un DSL en el desarrollo de proyectos software, así como también expresa que existen varios inconvenientes mostrados en las siguientes tablas:

**Tabla 1-2:** Ventajas de los DSL

Característica	Definición
<b>Mejora de la productividad del desarrollo</b>	El corazón del atractivo de un DSL es que proporciona un medio para comunicar más claramente la intención de una parte de un sistema. Si lees la definición de controlador de la señorita Grant en un formulario DSL, es más fácil para ti entender lo que está haciendo que a través de la API de consulta de comandos del modelo.
<b>Comunicación con los expertos en dominios</b>	Creo que la parte más difícil de los proyectos de software, la fuente más común de fracaso de los proyectos es la comunicación con los clientes y usuarios de ese software. Al proporcionar un lenguaje claro y preciso para tratar los dominios, un DSL puede ayudar a mejorar esta comunicación.
<b>Cambio en el contexto de ejecución</b>	Al hablar de por qué podríamos querer expresar nuestra máquina de estados en XML, una razón de peso era que la definición podía evaluarse en tiempo de ejecución en lugar de en tiempo de compilación. Este tipo de razonamiento, en el que queremos que el código se ejecute en un entorno diferente, es un impulsor común para el uso de un DSL. En el caso de los archivos de configuración de XML, el cambio de la lógica del tiempo de compilación al tiempo de ejecución es una razón común.

**Tabla 2-2:** Ventajas de los DSL (Continuación de la tabla anterior)

<b>Característica</b>	<b>Definición</b>
<b>Modelo Computacional Alternativo</b>	La programación de la corriente principal se hace más o menos usando un modelo de computación imperativo. Esto significa que le decimos a la computadora qué cosas hacer en qué secuencia, el flujo de control se maneja usando condicionales y bucles, tenemos variables - de hecho, muchas cosas que damos por sentado. La computación imperativa se ha hecho popular porque es relativamente fácil de entender y fácil de aplicar a muchos problemas. Sin embargo, no siempre es la mejor opción.

**Fuente:** (Fowler, 2010, pp. 33–36)

**Realizado por:** Núñez, Gilberth, 2021

**Tabla 3-2:** Desventajas de los DSL

<b>Característica</b>	<b>Definición</b>
<b>Lenguaje Cacofonía</b>	la preocupación de que los idiomas son difíciles de aprender, por lo que usar muchos idiomas será mucho más complicado que usar uno solo. El hecho de tener que conocer varios idiomas hace que sea más difícil trabajar en el sistema e introducir nuevas personas en el proyecto.
<b>El costo de la construcción</b>	Un DSL puede ser un pequeño costo incremental sobre su biblioteca subyacente, pero sigue siendo un costo. Todavía hay código que escribir, y sobre todo que mantener. Por lo tanto, como cualquier código, tiene que tirar de su peso. No todas las bibliotecas se benefician de tener un envoltorio DSL sobre él. Si una API de consulta de comandos hace el trabajo bien, entonces no tiene sentido añadir otra API encima. Incluso si un DSL puede ayudar, a veces sería demasiado esfuerzo construir y mantener para el beneficio marginal
<b>Lenguaje del gueto</b>	El problema del lenguaje del gueto contrasta con el problema de la cacofonía del lenguaje. Aquí, tenemos una compañía que ha construido muchos de sus sistemas en un lenguaje interno que no se utiliza en ningún otro lugar. Esto hace que sea difícil para ellos encontrar nuevo personal y mantenerse al día con los cambios tecnológicos
<b>Abstracción con parpadeo</b>	La utilidad de un DSL es que proporciona una abstracción que se puede utilizar para pensar en un área temática. Tal abstracción es realmente valiosa; permite expresar el comportamiento de un dominio mucho más fácilmente que si se piensa en términos de construcciones de nivel inferior.

**Fuente:** (Fowler, 2010, pp. 36–39)

**Realizado por:** Núñez, Gilberth, 2021

### 2.1.3 Análisis

Los DSL aportan productividad en la empresa facilitan la comunicación entre desarrolladores y sus clientes, son utilizados comúnmente para solucionar problemas o dominios específicos como mencionan ambos autores y lo que resulta atractivo para los desarrolladores es la facilidad de uso en comparación con los GPL. Se optó por el desarrollo del DSL para mejorar la productividad en el desarrollo de sistemas que posean una arquitectura CRUD REST y aprovechar todas las ventajas que ofrece un DSL desarrollado

## 2.2 Python

### 2.2.1 Definición

Al hablar de Python nos referimos a un lenguaje de programación del cual habla Alberto Álvarez en su libro titulado “Programar con Python 3” y lo define como:

*Un lenguaje de propósito general, aplicándose en muchos campos de muy diferente naturaleza, en gran parte debido a su flexibilidad para incorporar código escrito en otros lenguajes y a unas bibliotecas muy potentes que le permiten extender sus capacidades fácilmente. Especialmente interesante es su crecimiento en el área científica, donde podemos encontrarlo en proyectos del más alto nivel (por ejemplo, en la NASA). (Cuevas, 2019, pp. 7–8).*

Otra definición nos da James Tudor la cual hace mención en su libro “Python para principiantes: Aprenda Python en 5 días con orientación paso a paso y ejercicios prácticos” y se refiere a Python como:

*Un lenguaje de programación interpretado de alto nivel para la programación de propósito general. Creado por Guido van Rossum, Python tiene una filosofía de diseño que enfatiza la legibilidad del código, lo que reduce el costo del mantenimiento del programa. No requiere compilación y, al igual que Java, es orientado a objetos. (Tudor, 2019)*

### 2.2.2 Características

En el libro de Raúl González menciona las características que posee Python en común con los lenguajes compilados los cuales se describen en la siguiente tabla

**Tabla 4-2:** Características de Python

<b>Característica</b>	<b>Descripción</b>
<b>Tipado dinámico</b>	La característica de tipado dinámico se refiere a que no es necesario declarar el tipo de dato que va a contener una determinada variable,
<b>Fuertemente tipado</b>	No se permite tratar a una variable como si fuera de un tipo distinto al que tiene, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente
<b>Multiplataforma</b>	El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.) por lo que si no utilizamos librerías específicas de cada plataforma nuestro programa podrá correr en todos estos sistemas sin grandes cambios.
<b>Orientado a objetos</b>	La orientación a objetos es un paradigma de programación en el que los conceptos del mundo real relevantes para nuestro problema se trasladan a clases y objetos en nuestro programa. Python también permite la programación imperativa, programación funcional y programación orientada a aspectos.

**Fuente:** (González, 2011, pp. 7-8)

**Realizado por:** Núñez, Gilberth, 2021

### 2.2.3 *Análisis*

Es un lenguaje de propósito general el cual enfatiza en la legibilidad del código lo cual reduce el costo del mantenimiento, es orientado a objetos, se aplica en campos de diferentes naturalezas por incorporar escritos en otros lenguajes y unas bibliotecas potentes que extienden sus capacidades fácilmente.

Se decidió usar el lenguaje Python por la popularidad alcanzada y la falta de herramientas que ayuden a mejorar la productividad de los desarrolladores

## 2.3 **Flask**

### 2.3.1 *Definición*

Según Shalabh Aggarwal en su libro titulado “Flask Framework Cookbook: Over 80 proven recipes and techniques for Python web development with Flask” en el cual se habla sobre técnicas para el desarrollo web usando Python con Flask y hace mención del concepto Flask refiriéndose a este como:

*Flask es un microframework ligero de aplicación web escrita en Python. Aprovecha la flexibilidad de Python para proporcionar una plantilla relativamente sencilla para el desarrollo de aplicaciones*

web. Flask permite escribir aplicaciones sencillas de una página, pero también tiene el poder de escalarlas y construir aplicaciones más grandes sin problemas. (Aggarwal, 2019, p. 1)

De la misma forma, se mencionan Copperwaite & Leifer en su libro llamado “Learning Flask Framework” se habla un poco del nacimiento de este framework y así mismo se refiere a flask como:

*Flask es un framework web ligero escrito en Python. Flask comenzó como una broma de April fool’s que se convirtió en un framework no muy popular en el mundo de los framework web en Python. Ahora es uno de los framework web Python más utilizados para start-ups, y se está convirtiendo, como la herramienta perfecta comúnmente aceptada para una solución rápida y simple en la mayoría de los negocios.* (Copperwaite & Leifer, 2015, p. 1)

### 2.3.2 Comparativa con otro framework

En su tesis, Alberto Sánchez(Sánchez, 2017, p. 14) realiza una tabla comparativa entre los frameworks utilizados comúnmente para el desarrollo de aplicaciones web en el lenguaje Python, la cual se presenta a continuación

**Tabla 5-2:** Comparativa entre las características de Django y Flask

<b>CARACTERISTICAS</b>	<b>DJANGO</b>	<b>FLASK</b>
<b>CATEGORIA</b>	Framework	Micro-framework
<b>BOOTSTRAPING</b>	SI (django-admin)	NO
<b>ORM</b>	PROPIO	SQLAlchemy
<b>ALCANCE PROYECTO</b>	MEDIO/GRANDE	PEQUEÑO
<b>CARACTERISTICAS</b>	Batteries include	FLEXIBILIDAD/SIMPLICIDAD
<b>CURVA DE APRENDIZAJE</b>	ALTA	BAJA
<b>SISTEMA DE PLANTILLAS</b>	PROPIO	JINJA
<b>I18N</b>	PROPIO	BABEL
<b>INTERNACIONALIZACION</b>	PROPIO	BLINKER
<b>MODELO MVC</b>	SI	NO

**Fuente:** (Sánchez, 2017, p. 14)Análisis

Para facilitar el desarrollo de la aplicación se optó por utilizar el microframework flask el cual es flexible y nos permite realizar proyectos con un control total, es utilizado también para el desarrollo de los DSL por la flexibilidad que presenta.

## **2.4 Pycharm**

### **2.4.1 Definición**

En el sitio oficial de JetBrains nos ofrecen un concepto del IDE PyCharm orientado a la consistencia en los sistemas operativos y las ediciones disponibles de este IDE:

*PyCharm es un IDE multiplataforma que proporciona una experiencia consistente en los sistemas operativos Windows, MacOS y Linux.*

*PyCharm está disponible en tres ediciones: Professional, Community y Edu. Las ediciones Community y Edu son proyectos de código abierto y son gratuitas, pero tienen menos características. PyCharm Edu ofrece cursos y ayuda a aprender a programar con Python. La edición Professional es comercial, y proporciona un conjunto excepcional de herramientas y características. (JetBrains s.r.o., 2020b)*

Según Quan Nguyen en su libro “Hands-On Application Development with PyCharm: Accelerate your Python applications using practical coding techniques in PyCharm” compara PyCharm con otros IDEs que llevan más tiempo como herramienta para los desarrolladores y una de las características es la interfaz de la cual se detalla:

*En términos de soporte, Spyder tiene una UI más simple (que se asemeja a la de MATLAB). Esto puede ser beneficioso para los programadores principiantes, así como para las personas que están pasando de MATLAB a Python. En cuanto a PyCharm, una interfaz completa con diferentes opciones de personalización van a ser intimidantes para los programadores no familiarizados, pero puede mejorar enormemente la productividad. Sin embargo, como son producidos por la misma empresa, los usuarios de IntelliJ IDEA específicamente tampoco tendrán problemas para cambiar a Python. (Nguyen, 2019, p. 17)*

### **2.4.2 Características**

En el sitio oficial de JetBrains se encuentran detalladas las características que este IDE ofrece las cuales se muestran en la siguiente tabla.



**Tabla 6-2:** Características de PyCharm

<b>CARACTERISTICAS</b>	<b>Descripción</b>
<b>Asistencia de codificación inteligente</b>	PyCharm proporciona finalización inteligente de códigos, inspecciones de códigos, resaltado de errores sobre la marcha y reparaciones rápidas, junto con refactorización automatizada de códigos y ricas capacidades de navegación.
<b>Editor de código inteligente</b>	El editor de código inteligente de PyCharm proporciona un soporte de primera clase para Python, JavaScript, CoffeeScript, TypeScript, CSS, lenguajes de plantillas populares y más. Aprovecha las ventajas de la finalización del código en el idioma, la detección de errores y las correcciones de código sobre la marcha.
<b>Navegación con código inteligente</b>	Utiliza la búsqueda inteligente para saltar a cualquier clase, archivo o símbolo, o incluso a cualquier ventana de acción o herramienta del IDE. Sólo se necesita un clic para cambiar a la declaración, super método, prueba, usos, implementación, y más.
<b>Refactorización rápida y segura</b>	Refactoriza tu código de forma inteligente, con seguro Renombrar y Borrar, Método de Extracción, Introducir Variable, Variable o Método en Línea, y otros refractores. El lenguaje y el marco específico de refactorización le ayudan a realizar cambios en todo el proyecto.
<b>Herramientas de desarrollo incorporadas</b>	La enorme colección de herramientas de PyCharm incluye un depurador integrado y un programa de pruebas; un perfilador Python; una terminal incorporada; integración con las principales herramientas de VCS y de bases de datos incorporadas; capacidades de desarrollo a distancia con intérpretes a distancia; una terminal ssh integrada; e integración con Docker y Vagrant.
<b>Depuración, pruebas y perfiles</b>	Utiliza el poderoso depurador con una interfaz gráfica para Python y JavaScript. Crea y ejecuta tus pruebas con ayuda de la codificación y un ejecutor de pruebas basado en la interfaz gráfica. Toma el control total de tu código con la integración de Python Profiler.
<b>Herramientas de la base de datos</b>	Accede a Oracle, SQL Server, PostgreSQL, MySQL y otras bases de datos directamente desde el IDE. Confíe en la ayuda de PyCharm para editar el código SQL, ejecutar consultas, navegar por los datos y alterar los esquemas.
<b>Desarrollo Web</b>	Además de Python, PyCharm proporciona un soporte de primera clase para varios marcos de desarrollo web Python, lenguajes de plantillas específicos, JavaScript, CoffeeScript, TypeScript, HTML/CSS, AngularJS, Node.js, y más.
<b>IDE personalizable y multiplataforma</b>	Utilice PyCharm en Windows, Mac OS y Linux con una sola clave de licencia. Disfrute de un espacio de trabajo ajustado con esquemas de color y combinaciones de teclas personalizables, con emulación VIM disponible.

Fuente: (JetBrains s.r.o., 2020a)

Realizado por: Núñez, Gilberth, 2021

### 2.4.3 Análisis

Este IDE en su edición *community* fue seleccionado para la creación de la aplicación modelo debido a que es de código abierto y gratuito, además de poseer todas las herramientas necesarias para el desarrollo de este proyecto como lo son la asistencia inteligente a la codificación el cual nos ayudara a minimizar los errores en sintaxis y el modo *debug* para encontrar algún error en la codificación.

## 2.5 Entornos Virtuales

### 2.5.1 Definición

Según Romano et al. en su libro habla sobre los entornos virtuales de Python y brevemente describe su funcionamiento al trabajar con distintas versiones de Python:

*Cada ambiente virtual que creamos con venv es un ambiente aislado y tendrá su propio conjunto independiente de paquetes Python instalados en sus directorios(carpetas) de sitios. Cuando creamos un entorno virtual con venv en Python 3.4 y superior, pip se incluye en el nuevo entorno virtual. En python 3.3, fue necesario instalar manualmente pip después de crear el entorno virtual.*  
(Romano et al., 2018, p. 368)

Los entornos virtuales resuelven varias situaciones y problemas que se encuentran en los entornos de desarrollos varios escenarios son planteados por Arturo Fernández en su libro el cual habla de esta solución:

*Existen para Python un conjunto de herramientas que permiten crear entornos aislados, donde cada uno de ellos es responsable de sus propios paquetes. Es decir, es posible crear una sandbox en la que instalar diferentes módulos que serán independientes, tanto de los que existen a nivel de sistema (directorio site-packages), como unos de otros. De esta forma, por ejemplo, para comprobar si un determinado modulo es compatible con nuestra aplicación, bastará con crear un entorno virtual, instalar todos los paquetes, incluida la nueva versión del paquete en cuestión. Si la aplicación funciona correctamente, entonces podremos instalar esta nueva versión en nuestro entorno de producción.*(FERNANDEZ, 2013a, p. 236)

### 2.5.2 Problemas que soluciona

En el sitio oficial de Virtualenv hablan de los problemas que soluciona usar un entorno virtual en el cual se presenta los siguientes escenarios.

**Tabla 7-2:** Problemas que soluciona un entorno virtual

<b>Problemas que soluciona un entorno virtual</b>
Imagina que tienes una aplicación que necesita la versión 1 de LibFoo, pero otra aplicación requiere la versión 2. ¿Cómo puedes usar ambas bibliotecas? Si instalas todo en tu python anfitrión (por ejemplo, python3.8) es fácil terminar en una situación en la que dos paquetes tienen requerimientos conflictivos.
¿qué pasa si quieres instalar una aplicación y dejarla como está? Si una aplicación funciona, cualquier cambio en sus librerías o en las versiones de esas librerías puede romper la aplicación
¿qué pasa si no puedes instalar los paquetes en el directorio global de sitios-paquetes, debido a que no tienes permisos para cambiar el entorno de python del host?

Fuente: (PyPA, 2007)

Realizado por: Núñez, Gilberth, 2021

### 2.5.3 Análisis

El uso de entornos virtuales en el proyecto nos permitió crear entornos aislados donde cada uno use las librerías que necesite para la ejecución y además nos permite controlar las versiones de esas librerías de manera que ambos proyectos sean independientes o si se desea un control en las librerías de cada modulo

## 2.6 PostgreSQL

### 2.6.1 Definición

Según el sitio oficial de PostgreSQL en su definición hace mención a la reputación adquirida por este a lo largo de los años y también a las características que posee de esa forma lo definen como “PostgreSQL es un poderoso sistema de base de datos relacional de objetos de código abierto con más de 30 años de desarrollo activo que le ha ganado una fuerte reputación por su fiabilidad, robustez de características y rendimiento” (PostgreSQL, 2020).

Un concepto similar lo abordan Regina y Leo en su libro “PostgreSQL: Up and Running” en el cual mencionan que:

*PostgreSQL tiene más objetos de base de datos que la mayoría de los otros productos de bases de datos relacionales (y eso es antes de los complementos). Probablemente nunca tocará muchos de estos objetos, pero si se le ocurre algo nuevo, lo más probable es que ya esté implementado usando uno de esos objetos esotéricos (Obe & Hsu, 2014, p. 4)*

### 2.6.2 Características

En el libro de Korry Douglas y Susan Douglas se habla de lo mucho que se ha beneficiado a PostgreSQL por su larga historia lo cual lo convierte en uno de los servidores más avanzados disponibles y de las características que se encuentran en una distribución estándar de PostgreSQL, las cuales se detallan en la siguiente tabla.

**Tabla 8-2:** Características de PostgreSQL

<b>CARACTERISTICAS</b>	<b>DEFINICION</b>
<b>OBJETO-RELACIONAL</b>	En PostgreSQL, cada tabla define una clase. PostgreSQL implementa la herencia entre tablas (o, si lo desea, entre clases). Las funciones y los operadores son polimórficos.
<b>CUMPLIMIENTO DE ESTÁNDARES</b>	La sintaxis de PostgreSQL implementa la mayoría del estándar SQL92 y muchas características de SQL99. Cuando se producen diferencias en la sintaxis, éstas se relacionan más a menudo con características exclusivas de PostgreSQL.
<b>CÓDIGO ABIERTO</b>	Un equipo internacional de desarrolladores mantiene PostgreSQL. Los miembros del equipo van y vienen, pero los miembros principales han estado mejorando el rendimiento y el conjunto de características de PostgreSQL desde por lo menos 1996.
<b>PROCESAMIENTO DE TRANSACCIONES</b>	PostgreSQL protege los datos y coordina a los múltiples usuarios actuales a través del procesamiento completo de transacciones. El modelo de transacción utilizado por PostgreSQL se basa en el control de concurrencia de múltiples versiones (MVCC). MVCC proporciona un rendimiento mucho mejor que el que se encontraría con otros productos que coordinan múltiples usuarios a través de un bloqueo a nivel de tabla, página o fila.
<b>INTEGRIDAD REFERENCIAL</b>	PostgreSQL implementa una completa integridad referencial al soportar relaciones clave externas y primarias, así como disparadores. Las reglas de negocio pueden ser expresadas dentro de la base de datos en lugar de depender de una herramienta externa.
<b>MÚLTIPLES LENGUAJES DE PROCEDIMIENTO</b>	Los disparadores y otros procedimientos pueden ser escritos en cualquiera de varios lenguajes de procedimiento. El código del lado del servidor se escribe más comúnmente en PL/pgSQL, un lenguaje de procedimientos similar al PL/SQL de Oracle. También se puede desarrollar código del lado del servidor en Tcl, Perl, incluso bash (el shell de código abierto Linux/Unix).

**Tabla 9-2:** Características de PostgreSQL (Continuación de la tabla anterior)

<b>CARACTERISTICAS</b>	<b>DEFINICION</b>
<b>MÚLTIPLES LENGUAJES DE PROCEDIMIENTO</b>	Los disparadores y otros procedimientos pueden ser escritos en cualquiera de varios lenguajes de procedimiento. El código del lado del servidor se escribe más comúnmente en PL/pgSQL, un lenguaje de procedimientos similar al PL/SQL de Oracle. También se puede desarrollar código del lado del servidor en Tcl, Perl, incluso bash (el shell de código abierto Linux/Unix).
<b>APIS PARA MÚLTIPLES CLIENTES</b>	PostgreSQL soporta el desarrollo de aplicaciones cliente en muchos lenguajes. Este libro describe cómo hacer interfaz con PostgreSQL desde C, C++, ODBC, Perl, PHP, Tcl/Tk, y Python.
<b>TIPOS DE DATOS ÚNICOS</b>	PostgreSQL proporciona una variedad de tipos de datos. Además de los habituales tipos numéricos, de cadena y de fecha, también encontrará tipos geométricos, un tipo de datos booleano y tipos de datos diseñados específicamente para tratar con direcciones de red.
<b>EXTENSIBILIDAD</b>	Una de las características más importantes de PostgreSQL es que puede ser extendido. Si no encuentras algo que necesitas, normalmente puedes añadirlo tú mismo. Por ejemplo, puede agregar nuevos tipos de datos, nuevas funciones y operadores, e incluso nuevos lenguajes de procedimiento y de cliente. Hay muchos paquetes contribuidos disponibles en Internet. Por ejemplo, Refractive Research, Inc. ha desarrollado un conjunto de tipos de datos geográficos que pueden ser utilizados para modelar eficientemente los datos espaciales (GIS).

**Fuente:** (Douglas & Douglas, 2003, pp. 1-2)

**Realizado por:** Núñez, Gilberth, 2021

### 2.6.3 *Análisis*

Se usó *Postgres* sistema de base de datos reputación por su fiabilidad, robustez de características y rendimiento, una de las características que se tomó en cuenta para la selección de este sistema fue ser de código abierto y esto permite una actualización constante para ofrecer un mejor servicio, así como también al ser de tipo objeto relacional podemos trabajar con herencia entre tablas y que tanto las funciones como operaciones sean polimórficas.

## 2.7 MongoDB

### 2.7.1 Definición

La definición de la que hablan los creadores de MongoDB en su sitio oficial hace referencia a la modernidad y la nueva era de desarrollo de la cual se dice que “MongoDB es una base de datos distribuida, basada en documentos y de uso general que ha sido diseñada para desarrolladores de aplicaciones modernas y para la era de la nube” (MongoDB, 2020)

Kristina Chodorow al hablar de MongoDB nos facilita en breves rasgos las características que posee MongoDB en su libro y se refiere como “MongoDB es una base de datos de propósito general poderosa, flexible y escalable. Combina la capacidad de escalar con características como índices secundarios, consultas de rango, clasificación, agregaciones e índices geoespaciales” (Chodorow, 2013, p. 3)

### 2.7.2 Características

Al hablar de las características de MongoDB Kristina Chodorow menciona que está destinada a ser una base de datos de uso general y que presenta una lista de características únicas las cuales se muestran en la siguiente tabla.

**Tabla 10-2:** Características de MongoDB

<b>CARACTERISTICAS</b>	<b>DEFINICION</b>
<b>INDEXACIÓN</b>	MongoDB soporta índices secundarios genéricos, permitiendo una variedad de consultas rápidas, y también proporciona capacidades de indexación únicas, compuestas, geoespaciales y de texto completo.
<b>AGREGACIÓN</b>	MongoDB soporta un "pipeline de agregación" que le permite construir agregaciones complejas a partir de piezas simples y permitir que la base de datos la optimice.
<b>TIPOS DE COLECCIONES ESPECIALES</b>	MongoDB apoya las colecciones de tiempo de vida para los datos que deben expirar en un momento determinado, como las sesiones. También admite colecciones de tamaño fijo, que son útiles para mantener datos recientes, como los registros.
<b>ALMACENAMIENTO DE ARCHIVOS</b>	MongoDB soporta un protocolo fácil de usar para almacenar archivos grandes y metadatos de archivos.

**Fuente:** (Chodorow, 2013, p. 4)

**Realizado por:** Núñez, Gilberth, 2021

### 2.7.3 Análisis

Para el desarrollo de una aplicación que use una base de datos no relacional se utilizó MongoDB que es el gestor de base de datos no relacional más popular y con varias características que nos facilita el acceso a los datos, aparte del poder de crear índices los cuales nos permitieron facilitar dicho acceso

## 2.8 Object-Relational mapping

### 2.8.1 Definición

En el libro de John Kouraklis habla sobre lo que es un ORM y sobre su evolución, como definición se refiere a un ORM como:

*El mapa relacional de objetos (ORM) representa un conjunto de técnicas de programación informática que intentan que los sistemas incompatibles cooperen, se comuniquen e intercambien información. Al mismo tiempo, intentan hacer más fácil la vida de los desarrolladores.*

*Los sistemas en discusión son sistemas de bases de datos y sistemas que evolucionan a partir del paradigma dominante de la programación orientada a objetos (OOP). Las bases de datos están diseñadas para almacenar y proporcionar acceso a diversos tipos de datos de forma persistente. Los datos se almacenan en las bases de datos y sobreviven a la ejecución de las aplicaciones que los utilizan. Por otro lado, los desarrolladores que siguen los principios de la OOP piensan de manera muy diferente cuando se trata de la representación de datos en sus aplicaciones. (Kouraklis, 2019, p. 1)*

De la misma forma, Vijay Mehta en su libro da un concepto breve de lo que es un ORM y lo define como “El ORM es una forma automatizada de conectar un modelo de objeto, a veces denominado modelo de dominio, a una base de datos relacional mediante el uso de metadatos como descriptor del objeto y los datos” (Mehta, 2008, p. 4)

### 2.8.2 Facilidades de un ORM.

En el libro de Santosh Kumar habla sobre las principales características de un ORM las cuales se presentan a continuación

**Tabla 11-2:** Facilidades que proporciona un ORM

Facilidades que proporciona ORM
Una API de alto nivel para crear, leer, actualizar y eliminar los objetos de persistencia.
Un lenguaje de consulta orientado a objetos para consultar los objetos de persistencia.
Un formato de metadatos para asignar los objetos de persistencia a elementos relacionales.
Un mecanismo de caché con modos de bloqueo adecuados.
Un soporte para mapear el modelo de objeto de dominio complejo al modelo de base de datos.

**Fuente:** (Kumar, 2009, p. 146)

**Realizado por:** Núñez, Gilberth, 2021

### 2.8.3 *Análisis*

Un ORM nos permitió que las bases de datos y los sistemas cooperen a través del paradigma de la programación orientada a objetos, lo cual nos permitió ir diseñando las bases de datos a partir de un esquema, pero si este necesita una modificación, sin que esto no desencadenen en varios procesos.

## 2.9 **Psycopg2**

### 2.9.1 *Definición*

En su libro Juba y Volkov expresan que psycopg2 es la librería más comúnmente usado por los desarrolladores al trabajar en Python con acceso a una base de datos Postgress y expresan que:

*Psycopg2 es uno de los controladores PostgreSQL más populares para Python. Cumple con la especificación DB API 2.0 de Python. Está implementado en su mayor parte en C y utiliza la biblioteca, libpq. Es seguro para los hilos, lo que significa que puedes compartir el mismo objeto de conexión entre varios hilos. Puede funcionar tanto con Python 2 como con Python 3. (Juba & Volkov, 2017, p. 408)*

En su libro Di Gregorio et al. expresan que este driver es popular en la base de datos PostgreSQL para el lenguaje de programación Python y hablan sobre las características de esta librería.

*Psycopg2 se implementa mayormente en C como una envoltura de libpq, resultando ser tanto eficiente como segura. Tiene cursores del lado del cliente y del lado del servidor, comunicación asíncrona y notificaciones, soporte de COPIA. Muchos tipos de Python están soportados desde el principio y adaptados a los tipos de datos PostgreSQL; la adaptación se puede ampliar y personalizar gracias a un sistema flexible de adaptación de objetos. (Di Gregorio et al., 2020)*



## 2.9.2 Análisis

Psycopg2 nos permitió que tanto la base de datos relacional como lo es *Postgres* y nuestro sistema cooperen a través del ORM *SQLAlchemy*.

## 2.10 SQLAlchemy

### 2.10.1 Definición

Es importante conocer de primera fuente como conceptualizan la herramienta es así como el sitio web oficial de *SQLAlchemy* nos ofrece la definición tal cual:

*SQLAlchemy es el kit de herramientas de Python SQL y Object Relational Mapper que da a los desarrolladores de aplicaciones todo el poder y la flexibilidad de SQL. Proporciona un conjunto completo de patrones de persistencia de nivel empresarial bien conocidos, diseñados para un acceso eficiente y de alto rendimiento a la base de datos, adaptados a un lenguaje de dominio simple y Python (SQLAlchemy, 2020).*

Un concepto similar que enfatiza lo relacionado con el mapeo que apoya la definición anterior es de la que habla Rick Copeland en su libro “Essential SQLAlchemy” en la cual se refiere como:

*SQLAlchemy es la Biblioteca Python creada por Mike Bayer para proporcionar una interfaz pitónica (idiomáticamente Python) de alto nivel a bases de datos relacionales como Oracle, DB2, MySQL, PostgreSQL y SQLite. SQLAlchemy intenta ser discreto con su código Python, permitiéndole mapear objetos Python simples y viejos (POPOs) a tablas de bases de datos sin cambiar sustancialmente su código Python existente. (Copeland, 2008, p. 1)*

### 2.10.2 Características

En su página Covantec habla sobre las características que ofrece el kit de herramientas *SQLAlchemy* la cual ofrece a los desarrolladores de aplicaciones flexibilidad y potencia de SQL, las características a las que se refiere se muestran en la siguiente tabla.

**Tabla 12-2:** Características de SQLAlchemy

<b>Característica</b>	<b>Descripción</b>
<b>Un ORM de potencia industrial</b>	Permiten la persistencia transparente de objetos utilizando un sistema de configuración declarativo
<b>Sistema de consulta orientado a la relación</b>	Expone explícitamente toda la gama de capacidades de SQL, incluidas combinaciones, subconsultas, correlaciones y casi todo lo demás, en términos del modelo de objetos
<b>Sistema completo y flexible</b>	Las colecciones se almacenan en caché dentro de una sesión y se pueden cargar en un acceso individual
<b>Sistema de construcción Core SQL y una capa de interacción DBAPI.</b>	SQLAlchemy Core es independiente del ORM y es una capa de abstracción de base de datos completa por derecho propio
<b>Base de datos de introspección y generación</b>	Los esquemas de la base de datos se pueden «reflejar» en un solo paso en las estructuras de Python que representan los metadatos de la base de datos

**Fuente:** (Covantec R.L, 2019)

**Realizado por:** Núñez, Gilberth, 2021

### 2.10.3 Análisis

Para facilitar la cooperación entre la base de datos relacional y el sistema se utilizó el kit de herramientas de *SQLAlchemy* el cual permitió la comunicación del sistema con una base relacional en este caso *Postgres*

## 2.11 MongoAlchemy

### 2.11.1 Definición

En el sitio oficial de PyMongo nos habla sobre las herramientas que han sido escritas para trabajar con él y en el apartado de capas similares a las ORMs hablan de MongoAlchemy como “otra capa parecida a ORM en la parte superior de PyMongo. Su API está inspirado en SQLAlchemy” (MongoDB, 2014)

En el libro de Gareth Dwyer habla sobre las extensiones que trabajan con Flask y MongoDB el cual habla sobre Flask-MongoAlchemy y expresa que es “quizás la más similar a SQLAlchemy (y no sólo por su nombre) es Flask-MongoAlchemy. Similar a SQLAlchemy, MongoAlchemy no es específica de Flask” (Dwyer, 2016, p. 236)

### 2.11.2 Análisis

Para facilitar la cooperación entre la base de datos no relacional y el sistema se utilizó el kit de herramientas de MongoAlchemy el cual permitió la comunicación del sistema con MongoDB

## 2.12 REST

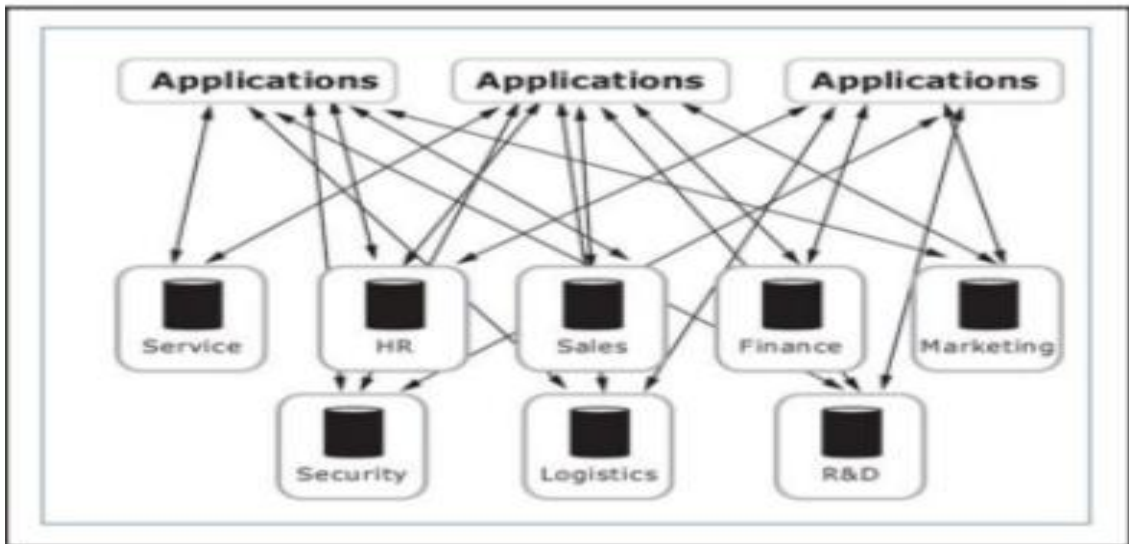
### 2.12.1 Definición

En su libro Dominic Duggan habla sobre el concepto de REST y su origen del cual expresa que:

*Los servicios Web REST surgieron en gran medida en oposición a los servicios Web basados en SOAP, aunque el concepto de REST se remonta al diseño de HTTP y la propia Web. REST no es un concepto difícil de entender, pero tiene cierta sutileza. Parte del desafío de apreciar las sutilezas de REST se debe sin duda a la "interferencia" de las concepciones preexistentes sobre las aplicaciones empresariales y distribuidas. Algunas de las filosofías de REST cuestionan la idoneidad de la RPC para las aplicaciones interempresariales, y abogan por un enfoque más flexible basado en la comunicación mediante recursos actualizables que tienen análogos en la vida real en los documentos comerciales que se intercambian como parte de las colaboraciones comerciales normales. Hay capas que pueden discernirse en la aplicación de los principios REST; cualquier arquitecto o desarrollador debería considerar cuidadosamente el nivel de aplicación de los principios REST que sea más apropiado a su situación (Duggan, 2012, p. 359)*

En su libro Bogunuva Balachandar habla sobre REST el cual lo define como:

*REST no es una arquitectura; más bien es un conjunto de restricciones que crea un estilo de arquitectura de software, que puede utilizarse para construir aplicaciones distribuidas. Uno de los principales problemas de las aplicaciones distribuidas se atribuye a la diversidad de los sistemas de una empresa que ofrecen silos de información comercial (Balachandar, 2017, p. 7)*



**Figura 2-2:** Representación de sistemas de una empresa que ofrecen silos de información

Fuente: (Balachandar, 2017, p. 7)

### 2.12.2 Restricciones de un sistema REST

En su libro Bogunuva Balachandar habla sobre REST y expresa que hay varias restricciones que definen a este estilo de arquitectura y dichas restricciones se muestran en la siguiente tabla

**Tabla 13-2:** Restricciones que definen un sistema REST

RESTRICCIÓN	DESCRIPCIÓN
<b>CLIENTE-SERVIDOR</b>	Esta restricción mantiene al cliente y al servidor vagamente acoplados. En este caso, el cliente no necesita conocer los detalles de implementación en el servidor, y el servidor no se preocupa de cómo los datos son utilizados por el cliente. Sin embargo, se mantiene una interfaz común entre el cliente y el servidor para facilitar la comunicación.
<b>SIN ESTADO</b>	No debería haber necesidad de que el servicio mantenga las sesiones de usuario. En otras palabras, cada solicitud debe ser independiente de las demás. Esto mejora la escalabilidad, ya que el servidor no necesita gestionar el estado en múltiples solicitudes, con algún compromiso en el rendimiento de la red.
<b>CACHEABLE</b>	Esta restricción tiene que soportar un sistema de caché. La infraestructura de la red debe soportar un caché a diferentes niveles. El cacheo puede evitar repetidos viajes de ida y vuelta entre el cliente y el servidor para recuperar el mismo recurso.

**Tabla 14-2:** Restricciones que definen un sistema REST (Continuación de la tabla anterior)

<b>RESTRICCIÓN</b>	<b>DESCRIPCIÓN</b>
<b>INTERFAZ UNIFORME</b>	Esta restricción indica una interfaz genérica para gestionar todas las interacciones entre el cliente y el servidor de forma unificada, lo que simplifica y desacopla la arquitectura. Esta restricción indica que cada recurso expuesto para su uso por el cliente debe tener una dirección única y debe ser accesible a través de una interfaz genérica. El cliente puede actuar sobre los recursos utilizando un conjunto genérico de métodos.
<b>SISTEMA DE CAPAS</b>	El servidor puede tener múltiples capas para su implementación. Esta arquitectura en capas ayuda a mejorar la escalabilidad al permitir el equilibrio de la carga. También mejora el rendimiento al proporcionar cachés compartidos en diferentes niveles. Siendo la puerta del sistema, la capa superior también puede hacer cumplir las políticas de seguridad.
<b>CÓDIGO BAJO DEMANDA</b>	Esta restricción es opcional. Esta restricción indica que la funcionalidad de las aplicaciones cliente se puede ampliar en tiempo de ejecución permitiendo una descarga de código del servidor y ejecutando el código. Algunos ejemplos son los applets y el código JavaScript que se transfieren y ejecutan en el lado del cliente en tiempo de ejecución.

**Fuente:** (Balachandar, 2017, p. 8)

**Realizado por:** Núñez, Gilberth, 2021

### **2.12.3 Análisis**

Para desacoplar el cliente del servidor se utilizó el estilo de arquitectura REST puesto que nos brinda un enfoque más flexible basado en la comunicación mediante recursos actualizables, lo cual nos permitió crear aplicaciones distribuidas

## **2.13 Restful**

### **2.13.1 Definición**

Al hablar de los servicios web RESTful podemos referirnos a una breve descripción que nos ofrece Bogunva Balachandar el cual lo define como “Los servicios web que se adhieren a las restricciones arquitectónicas de REST se caracterizan como servicios web RESTful” (Balachandar, 2017, p. 20)

En su libro Haafiz Ahmad habla sobre las restricciones que se debe cumplir para definir un sistema de tipo REST y también se refiere a los servicios web RESTful y los define como:

Los servicios web de RESTful se basan en los recursos de RESTful. Un recurso RESTful es una entidad/recurso que se almacena en su mayor parte en un servidor y que el cliente solicita utilizando los servicios web RESTful. Aquí hay algunas características de un recurso en términos de servicios web RESTful:

Es una entidad normalmente referida como un sustantivo en la URL, es única, tiene datos asociados a ella, tiene al menos una URI (Ahmad, 2017, p. 7)

### 2.13.2 Diferencia entre los servicios Web

En el libro de Tharam Dillon habla sobre las diferencias entre los servicios web SOAP y RESTful lo cual se detalla en la siguiente tabla

**Tabla 15-2:** Diferencia entre los servicios Web RESTful y SOAP

<b>Diferencia entre los servicios Web</b>	
<b>RESTful</b>	<b>SOAP</b>
No hay una descripción formal de los puntos finales del servicio (por ejemplo, no hay WSDL) entre el productor del servicio y el consumidor. Los productores de servicios suelen proporcionar algunos juegos de herramientas adicionales que describen cómo invocar los servicios web REST	Se requiere un contrato formal entre el proveedor de servicios y el consumidor. El lenguaje de descripción de servicios web (WSDL) se encarga de toda la información necesaria sobre las operaciones, los parámetros, los tipos y el formato de los mensajes de los servicios web. Esto es útil en situaciones en las que se requiere la composición del servicio en modelos de procesos comerciales complejos
La arquitectura tiene que manejar requisitos no funcionales como la escalabilidad, el ancho de banda de la red y la integración.	La arquitectura necesita manejar requerimientos complejos no funcionales como la seguridad, la confianza y la coordinación.
Arquitecturas en las que los servicios web son completamente asíncronas al igual que las peticiones HTTP en un sitio web. Por ejemplo, las aplicaciones web existentes pueden exponer su funcionalidad fácilmente a través de operaciones CRUD utilizando servicios web REST.	La arquitectura necesita manejar la invocación y el procesamiento asíncronico de las solicitudes de servicio.

Fuente: (Dillon, 2015, p. 131)

Realizado por: Núñez, Gilberth, 2021

### 2.13.3 Análisis

El servicio que se generará será de tipo RESTful dado que cumplen con todas las restricciones REST y que la mayor parte del recurso se almacena en el servidor el cual es solicitado por el cliente.

## 2.14 Serializadores

### 2.14.1 Definición

En el sitio oficial de *marshmallow* una librería que nos permite convertir complejos tipos de datos en datos nativos de Python nos habla sobre la serialización la cual lo define como “Serializar los objetos del nivel de aplicación a los tipos primitivos de Python. Los objetos serializados pueden entonces ser renderizados a formatos estándar como JSON para su uso en una API HTTP” (Loria, 2020a)

En su libro Alex Martelli habla sobre la definición de los serializadores de los cuales se dice que:

*Python suministra un número de módulos que se ocupan de las operaciones I/O que serializan (guardan) objetos Python enteros a varios tipos de corrientes de bytes, y deserializan (cargan y recrean) objetos Python de vuelta de dichas corrientes. La serialización también se llama "marshaling" (Martelli, 2003, p. 224)*

### 2.14.2 Datos que pueden ser serializados con Python pickle

En su libro Arturo Fernández habla sobre los tipos de datos que pueden ser serializados los cuales se muestran en la siguiente tabla

**Tabla 16-2:** Datos que pueden ser serializados con Python pickle

<b>Datos que pueden ser serializados con pickle</b>
Aquellos que toman los valores True, False y None
Números enteros, complejos y reales
Cadenas de texto, bytes y listas de bytes
Funciones definidas en el nivel superior de un módulo
Funciones integradas en el intérprete que residan en el nivel superior de un módulo
Tuplas, listas, conjuntos y diccionarios que contengan elementos que pertenezcan a los tipos anteriores
Clases definidas por el programador que contengan elementos que pertenezcan a los tipos anteriores

Fuente: (FERNANDEZ, 2013b)

Realizado por: Núñez, Gilberth, 2021

### **2.14.3 Análisis**

Los serializadores nos permitió enviar los datos de forma serializada es decir a tipos corrientes de bytes los cuales al ser recibidos se des serializan a tipos de datos que puedan ser interpretados

## **2.15 Marshmallow**

### **2.15.1 Definición**

En el sitio oficial de marshmallow a esta librería se la define como “Marshmallow se inspira en el Marco REST de Django, Flask-RESTful, y colador. Toma prestadas varias ideas de implementación y diseño de estas bibliotecas para crear una solución flexible y productiva para la clasificación, desclasificación y validación de datos” (Loria, 2020b)

En su libro Gaston Hillar habla sobre la librería marshmallow y se refiere a este como:

*Marshmallow es una biblioteca ligera para convertir tipos de datos complejos a y desde tipos de datos nativos de Python. Marshmallow proporciona esquemas que podemos utilizar para validar los datos de entrada, deserializar los datos de entrada a objetos de nivel de aplicación, y serializar los objetos de nivel de aplicación a tipos primitivos de Python (Hillar, 2018, p. 63)*

### **2.15.2 Análisis**

Esta librería nos permitió realizar el proceso de serialización para enviar los datos solicitados por un cliente de forma efectiva y segura.

## **2.16 JSON**

### **2.16.1 Definición**

En su libro BEN SMITH habla sobre la definición de JSON como un formato de intercambios de objetos al cual lo describe como:

*El formato de datos de la Notación de Objetos de JavaScript, o JSON para abreviar, se deriva de los literales del lenguaje de programación JavaScript. Esto hace que JSON sea un subconjunto del lenguaje JavaScript. Como subconjunto, JSON no posee ninguna característica adicional que el propio lenguaje JavaScript no posea ya. Aunque JSON es un subconjunto de un lenguaje de programación, no es en sí mismo un lenguaje de programación sino, de hecho, un formato de intercambio de datos. JSON se conoce como el estándar de intercambio de datos, lo que implica subtextualmente que puede utilizarse como formato de datos dondequiera que se produzca el intercambio de datos. Un intercambio de datos puede ocurrir entre un navegador y un servidor e*



*incluso de servidor a servidor, para el caso. Por supuesto, estos no son los únicos medios posibles para intercambiar JSON, y dejarlo en esos dos sería más bien limitante (SMITH, 2015, p. 37)*

En su libro Sai Sriparasa y Bruno D'mello hablan de JSON como un formato de intercambio de datos el cual es independiente del lenguaje, lo definen como:

*Para definir JSON, podemos decir que es un formato basado en texto, ligero y legible por el hombre para el intercambio de datos entre clientes y servidores. JSON se deriva de JavaScript y se asemeja mucho a los objetos de JavaScript, pero no depende de JavaScript. JSON es independiente del lenguaje, y el soporte para el formato de datos JSON está disponible en todos los lenguajes populares, algunos de los cuales son C#, PHP, Java, C++, Python y Ruby (Sriparasa & D'mello, 2018, p. 6)*

### 2.16.2 Comparativa con XML

En su libro Dixit habla sobre las similitudes y diferencias que existen entre ambos formatos de intercambio de datos lo cual se muestra en la siguiente tabla

**Tabla 17-2:** Comparativa con XML

<b>Similitud</b>	<b>Diferencias</b>
JSON es un texto simple que sirve para almacenar e intercambiar información de texto.	JSON es más corto/pequeño que XML
JSON es "autodescriptivo" (legible para los humanos).	JSON es más fácil de analizar y más rápido de leer y escribir.
JSON es jerárquico (valores dentro de valores).	JSON debe ser analizado usando JavaScript eval()
JSON puede ser analizado por JavaScript.	JSON utiliza matrices
Los datos de JSON pueden ser transportados usando AJAX.	JSON no tiene palabras reservadas.
	JSON no tiene etiquetas finales como XML

**Fuente:** (Dixit, 2014, pp. 230–231)

**Realizado por:** Núñez, Gilberth, 2021

### 2.16.3 Análisis

El formato de datos que será enviado al cliente que realice la petición será de tipo JSON ya que nos permite intercambiar datos en cualquier tipo de lenguaje, aparte es ligero, legible y puede ser interpretado por cualquier lenguaje de programación.

## 2.17 BSON

### 2.17.1 Definición

Bson es fácil de entender e indexar rápidamente de esto se habla en el libro de Hows et al. el cual define a BSON como:

*En resumen, BSON es mucho más fácil de atravesar (es decir, de mirar a través) e indexar muy rápidamente. Aunque BSON requiere un poco más de espacio en disco que JSON, este espacio extra es poco probable que sea un problema, porque los discos son baratos, y MongoDB puede escalar a través de las máquinas. La compensación en este caso es bastante razonable: se intercambia un poco de espacio extra en el disco para un mejor rendimiento de consulta e indexación. El segundo beneficio clave de usar BSON es que es fácil y rápido convertir BSON al formato de datos nativo de un lenguaje de programación. Si los datos se almacenaran en JSON puro, se necesitaría una conversión de relativamente alto nivel. Existen controladores de MongoDB para un gran número de lenguajes de programación (como Python, Ruby, PHP, C, C++ y C#), y cada uno funciona de manera ligeramente diferente. Utilizando un simple formato binario, se pueden construir rápidamente estructuras de datos nativas para cada lenguaje, sin necesidad de procesar primero JSON. Esto hace que el código sea más simple y rápido, y ambos están en consonancia con los objetivos declarados de MongoDB. BSON también proporciona algunas extensiones para JSON. Por ejemplo, permite almacenar datos binarios e incorporar un tipo de datos específico. Así, mientras que BSON puede almacenar cualquier documento JSON, un documento BSON válido puede no ser un JSON válido. Esto no importa, porque cada lenguaje tiene su propio controlador que convierte los datos desde y hacia BSON sin necesidad de usar JSON como lenguaje intermediario. (Hows et al., 2013, p. 11)*

En su libro Kristina Chodorow habla sobre el formato de intercambio de datos BSON a el cual lo define como:

*BSON es un formato binario ligero capaz de representar cualquier documento de MongoDB como una cadena de bytes. La base de datos entiende BSON, y BSON es el formato en el que los documentos se guardan en el disco.*

*Cuando se le da a un controlador un documento para insertar, utilizarlo como una consulta, etc., codificará ese documento a BSON antes de enviarlo al servidor. Del mismo modo, los documentos que se devuelven al cliente desde el servidor se envían como cadenas BSON. Estos datos BSON son decodificados por el controlador en su representación de documento nativo antes de ser devueltos al cliente. (Chodorow, 2013, p. 389)*

### 2.17.2 *Objetivos principales*

En su libro Kristina Chodorow habla sobre los objetivos principales de BSON los cuales se muestran en la siguiente tabla

**Tabla 18-2:** Objetivos principales de BSON

<b>Objetivo</b>	<b>Definición</b>
<b>Eficiencia</b>	BSON está diseñado para representar datos de forma eficiente, sin usar mucho espacio extra. En el peor de los casos, BSON es ligeramente menos eficiente que JSON; y en el mejor de los casos (por ejemplo, cuando se almacenan datos binarios o numéricos grandes), es mucho más eficiente.
<b>Traversabilidad</b>	En algunos casos, BSON sacrifica la eficiencia espacial para que el formato sea más fácil de atravesar. Por ejemplo, los valores de las cadenas se prefijan con una longitud en lugar de depender de un terminador para significar el final de una cadena. Esta capacidad de desplazamiento es útil cuando el servidor de MongoDB necesita introspectar documentos.
<b>Rendimiento</b>	Por último, BSON está diseñado para ser rápido para codificar y decodificar. Utiliza representaciones de estilo C para los tipos, que son rápidas de trabajar en la mayoría de los lenguajes de programación.

**Fuente:** (Chodorow, 2013, pp. 389–390)

**Realizado por:** Núñez, Gilberth, 2021

### 2.17.3 *Análisis*

BSON es un formato binario ligero que nos permitió trabajar con tipos de datos complejos en el desarrollo del sistema, ya que presenta facilidades el momento de enviar tipos de datos que son complicados el momento de armar un formato de tipo como JSON

## **2.18 Eclipse IDE**

### **2.18.1 Definición**

En su libro Ram Kulkarni habla sobre la definición del IDE Eclipse en el cual se habla de su mejora en la productividad en el desarrollo y varias características:

*Un buen IDE es esencial para una mejor productividad en la codificación. Eclipse es uno de esos IDE, que tiene grandes características de editor y muchos puntos de integración con las tecnologías JEE. El propósito principal de este libro es mostrarle cómo desarrollar aplicaciones JEE utilizando Eclipse. Por lo tanto, lo que sigue es una rápida introducción a Eclipse, si no está ya familiarizado con él.*

*Eclipse es un IDE de código abierto para desarrollar aplicaciones en muchos lenguajes de programación diferentes. Es bastante popular para desarrollar muchos tipos diferentes de aplicaciones Java. Su arquitectura es enchufable: hay un componente central del IDE y se le pueden añadir muchos plugins diferentes. De hecho, el soporte para muchos lenguajes se agrega como plugins de Eclipse, incluyendo el soporte para Java.*

*Junto con el soporte de editor, Eclipse tiene plugins para interactuar con muchos de los sistemas externos utilizados durante el desarrollo. Los ejemplos incluyen sistemas de control de fuentes como SVN y Git, herramientas de construcción como Apache Ant y Maven, exploradores de archivos para sistemas remotos que utilizan FTP, servidores de gestión como Tomcat y GlassFish, exploradores de bases de datos, perfiles de memoria y CPU. Veremos muchas de estas características en los capítulos siguientes. La siguiente captura de pantalla muestra la vista por defecto de Eclipse para el desarrollo de la aplicación TEE (Kulkarni, 2018, p. 13)*

Otra breve descripción sobre el IDE Eclipse nos ofrece Jeff Heaton el cual en su libro lo define como:

*Eclipse es un avanzado y muy popular IDE para la plataforma Java. Hay algunas especulaciones sobre de dónde viene el nombre de Eclipse. Algunos dicen que IBM quiso decir que "eclipsará" a Sun Microsystems en el mercado de herramientas de Java. Cualquiera que sea el significado detrás del nombre, es una herramienta valiosa. (Heaton, 2003, p. 345)*

### **2.18.2 Características**

En su libro Ed Burnette habla sobre las características de Python las cuales se describen en la siguiente tabla

**Tabla 19-2:** Características de Python

<b>Característica</b>	<b>Definición</b>
<b>Asistencia de código</b>	El editor de Java siempre está atento a lo que escribes, listo para ofrecer sugerencias útiles a través de una función llamada asistencia de código (también llamada asistencia de contenido).
<b>Plantillas</b>	Eclipse proporciona una forma abreviada de introducir texto llamada plantillas. Por ejemplo, en el editor de Java, si escribes para y presiona Ctrl + Espacio, la ventana de asistencia de código aparecerá.
<b>Escritura automática</b>	Estrechamente relacionada con la asistencia de código está una característica llamada escritura automática.
<b>Refactorización</b>	Refactorizar significa transformar el código sin cambiar su funcionalidad. Considere el cambio de nombre, que es la forma más simple de refactorización. Si renuncias a una variable local de rosa a margarita, olería igual de dulce.
<b>Ayuda con el cursor</b>	Has visto que la asistencia de código es una buena manera de explorar una API desconocida. Otra herramienta útil es la ayuda del cursor. Para usar la ayuda del cursor, simplemente mueve el cursor del ratón sobre un símbolo del que quieras saber más y haz una pausa por un momento.
<b>Hipervínculos</b>	¿Sabías que hay un navegador web incorporado en el editor de Java? Bueno, hay algo así. El editor te permite navegar por tu programa como si fuera un sitio web. Mantén pulsada la tecla Ctrl y mueve el ratón a través del código fuente. Aparecerá un subrayado para indicar los símbolos de los hipervínculos. Puede dejar el cursor del ratón sobre el símbolo para ver su definición, o hacer clic en él para abrir la declaración en el editor.
<b>Arreglos rápidos</b>	Cada vez que comete un error de sintaxis en su programa, el compilador de fondo de Eclipse lo detecta inmediatamente y dibuja un indicador de error (conocido cariñosamente como el garabato rojo) bajo el código ofensivo. Además de detectar simplemente el problema, Eclipse puede ofrecer normalmente una corrección automática del programa, llamada una solución rápida.

**Fuente:** (Burnette, 2005, pp. 38–43)

**Realizado por:** Núñez, Gilberth, 2021

### 2.18.3 Análisis

Se optó por el IDE Eclipse dado que posee un soporte para *frameworks* que nos permitan desarrollar el DSL y aprovechar las características que ofrece para mejorar la productividad al utilizar nuestro DSL, características como la asistencia de código nos apoyara en uso de nuestro DSL propuesto.

## 2.19 XText

### 2.19.1 Definición

Xtext es un *framework* desarrollado en el proyecto Eclipse, este framework es de código abierto orientado al desarrollo de lenguajes específicos de dominio (DSL) y lenguajes de programación.

*Xtext es un marco para el desarrollo de lenguajes de programación y de lenguajes específicos de dominios. Con Xtext se define el lenguaje utilizando un poderoso lenguaje gramatical. Como resultado, se obtendrá una completa infraestructura, que incluye analizador, enlazador, corrector tipográfico, compilador, así como soporte de edición para Eclipse, cualquier editor que soporte el Protocolo de Servidor de Lenguaje y un navegador web (Eclipse Foundation, 2020b).*

Según Lorenzo Bettini en su libro titulado “Implementing Domain-Specific Languages with Xtext and Xtend” habla sobre lo que es xtext y se expresa de la siguiente manera:

*Xtext es un framework de Eclipse para implementar lenguajes de programación y DSLs. Permite implementar lenguajes rápidamente y, sobre todo, cubre todos los aspectos de una infraestructura lingüística completa, desde el analizador, el generador de código o el intérprete, hasta una integración completa del IDE de Eclipse con todas las características típicas del IDE: ya lo hemos comentado anteriormente. Lo realmente sorprendente de Xtext es que, para iniciar una implementación de DSL, sólo necesita una especificación gramatical similar a la de ANTLR. No es necesario anotar las reglas con acciones para construir la AST, ya que la creación de la AST (y las clases de Java para almacenar la AST) es manejada automáticamente por el propio Xtext. A partir de esta especificación, Xtext generará automáticamente todos los mecanismos esbozados anteriormente. Generará el lexer, el parser, el modelo de AST, la construcción del AST para representar el programa analizado, y el editor de Eclipse con todas las características del IDE. Xtext viene con buenas e inteligentes implementaciones por defecto para todos estos aspectos, y de hecho la mayoría de estos valores por defecto seguramente se ajustarán a sus necesidades. Sin embargo, cada aspecto puede ser personalizado por el diseñador del lenguaje. Con todas estas características, Xtext es fácil de usar. Produce un resultado profesional rápidamente, e incluso es divertido de usar.*

*Desde la versión 2.9.0, Xtext le permite portar sin problemas su implementación de DSL y herramientas IDE a IntelliJ y también incrustar su editor DSL en una aplicación web. (Bettini, 2016, p. 12)*

### 2.19.2 Características

Las características que ofrece Xtext según Ángel López en su tesis “Lenguajes específicos de dominio gráficos y textuales: Un estudio comparativo” se detallan en la siguiente tabla.

**Tabla 20-2:** Características de Xtext

<b>Característica</b>	<b>Descripción</b>
<b>Coloración de sintaxis</b>	Ayuda a diferenciar las distintas partes como por ejemplo la búsqueda de errores en el código, así como también identificadores y palabras clave
<b>Auto inyección de texto</b>	Ofrece la posibilidad de introducir automáticamente valores a las propiedades de un componente
<b>Plantillas de opción múltiple</b>	Estas plantillas soportan a una resolución automática de referencias

Fuente: (López, 2013, p. 48)

Realizado por: Núñez, Gilberth, 2021

### 2.19.3 Análisis

Xtext es un framework de Eclipse orientado para el desarrollo de los lenguajes específicos de dominio que cubre todos los aspectos de la infraestructura lo cual lo hace muy completo y robusto para estos trabajos. Este framework fue seleccionado puesto que nos permite establecer la gramática y las reglas para el DSL propuesto, también poder apoyarnos en su completa infraestructura como el compilador y el corrector tipográfico para apoyar al desarrollador en el uso del DSL.

## 2.20 XTend

### 2.20.1 Definición

Eclipse Foundation en su sitio web oficial tiene un apartado dedicado sobre xtend en el cual lo definen como:

*Xtend es un dialecto flexible y expresivo de Java, que compila en código fuente legible compatible con Java 8. Puede usar cualquier biblioteca existente de Java sin problemas. La salida compilada es legible y bastante impresa, y tiende a correr tan rápido como el código Java equivalente escrito a mano* (Eclipse Foundation, 2020a)

Según Lorenzo Bettini en su libro titulado “Implementing Domain-Specific Languages with Xtext and Xtend” al hablar sobre xtend lo define de la siguiente manera:

*Usaremos Xtend a lo largo de este libro para escribir todas las partes de una implementación de DSL. Es decir, lo usaremos para personalizar las características de la interfaz de usuario, para*

*escribir pruebas, para implementar comprobaciones de restricciones y para escribir generadores o intérpretes de código para todos los DSL de ejemplo que desarrollaremos en este libro. En particular, todas las clases stub generadas por Xtext para sus proyectos DSL son clases Xtend. Aún puedes generar clases stub en Java personalizando el flujo de trabajo de MW E2, pero en este gancho, siempre usaremos clases Xtend. Xtend, además de proveer mecanismos útiles para escribir generadores de código, por ejemplo, expresiones de plantillas multilínea, también provee poderosas características que hacen que visitar y recorrer el modelo sea realmente fácil, directo y natural de leer y mantener. De hecho, además de la definición gramatical, para el resto del tiempo al implementar un DSL, tendrá que visitar el modelo AST. Los programas de Xtend están traducidos a Java, y el código de Xtend puede acceder a todas las bibliotecas de Java; así Xtend y Java pueden cooperar sin problemas. (Bettini, 2016, pp. 41–42)*

### 2.20.2 Ventajas y desventajas de Xtend

En la tesis de Luis Gonzales menciona que el uso de xtend presenta las siguientes ventajas y desventajas:

**Tabla 21-2:** Ventajas y desventajas de Xtend

Ventajas	Desventajas
<ul style="list-style-type: none"> <li>• Permite escribir código más conciso para plataformas Java.</li> <li>• Es compatible con muchas tecnologías Java</li> <li>• Como el lenguaje Xtend genera código fuente en Java y no código intermedio de Java, se pueden aplicar las optimizaciones del compilador al código generado.</li> </ul>	<ul style="list-style-type: none"> <li>• Fuerte acoplamiento con Eclipse, El código escrito en el lenguaje Xtend puede ser difícil su uso en otra herramienta que no sea Eclipse debido a que en principio fue desarrollado para Eclipse.</li> </ul>

**Fuente:** (Gonzales, 2015, pp. 10–11)

**Realizado por:** Núñez, Gilberth, 2021

### 2.20.3 Análisis

Xtend es un lenguaje de propósito general y puede ser utilizado de manera independiente, lo interesante de este lenguaje es que su salida compilada tiende a correr tan rápido como un código java equivalente. Xtend nos permitió interpretar la gramática del DSL propuesto y obtener como salida el código Python del proyecto



## **2.21 Productividad**

### **2.21.1 Definición**

La productividad es definida por David Rico en su libro denominado “ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers” como:

*La productividad es, en general, la cantidad de trabajo que se realiza. También es el número de productos y servicios creados. La productividad es el ritmo, la velocidad o la capacidad de un proceso de software. En este caso, la productividad es una medida de la rapidez y la cantidad de productos y servicios de software que se realizan en un periodo de tiempo determinado, que puede ser una hora, un día, un mes, un año, la duración de un proyecto o la media de una empresa productora de software. Por ejemplo, una productividad de 25 líneas de código por hora significa que por cada hora invertida se producen 25 líneas de código. (Rico, 2004, p. 42)*

Un concepto similar presenta Gómez et al. en su artículo titulado “Estudio del Proceso Software Personal (PSP) en un Entorno Académico” en el cual se habla de la productividad como:

*A nivel empresarial, la productividad es un indicador relevante para cuantificar los productos o servicios generados en términos del tiempo requerido para producirlos. En PSP la productividad se mide de acuerdo a las LOCs por hora (LOCs/Hora) que produce un ingeniero, siendo esta medición un indicador de la productividad individual (Gómez et al., 2014, p. 21)*

### **2.21.2 Análisis**

Para la evaluación del DSL se definió la característica de la productividad para la cual se tomó en cuenta el número de líneas de código y el tiempo requerido para producirlo. De esta forma verificar si con el uso del DSL se mejoró la productividad en el desarrollo de software.

## CAPÍTULO III

### 3 MARCO METODOLÓGICO

En este capítulo se presenta el tipo de investigación, métodos de investigación, técnicas de investigación, la fase de planificación, el *product backlog*, el *sprint backlog*, la planificación de la gestión de riesgos, los recursos necesarios, la fase de desarrollo, generación de la aplicación modelo y definición de la gramática, el desarrollo de los sprints, el estudio del dominio de la aplicación del DSL, la implementación de la aplicación modelo, la definición de la gramática del lenguaje, la generación del metamodelo, la implementación del generador de código, la ejemplificación de la propuesta del DSL, la gestión del proyecto y la gestión de los riesgos.

#### 3.1 Actividades de la metodología

##### 3.1.1 Tipo y métodos de investigación a utilizar

###### 3.1.1.1 Tipo de investigación

El tipo de estudio que se realizó para el desarrollo de este trabajo es la investigación aplicada e investigación descriptiva, la razón de ser una investigación aplicada se debe a la creación de un DSL en el cual se aplicó los conocimientos adquiridos en los años de estudio superior, al mismo tiempo se considera una investigación descriptiva dado que se detallan los procesos llevados a cabo para el desarrollo del presente trabajo de integración curricular.

###### 3.1.1.2 Métodos de investigación

#### Método Analítico-Sintético

En el análisis se estudió a las herramientas que nos permiten construir lenguajes específicos de dominio con el propósito de entender dichas herramientas para su uso en el desarrollo del DSL.

También se analizó proyectos desarrollados en el lenguaje *Python* con el micro-framework *Flask* para entender la estructura e identificar practicas comunes en el desarrollo de sistemas con el uso de estas herramientas es decir para realizar el estudio de dominio.

La síntesis se aplicó para el desarrollo de la aplicación modelo partiendo de la información recolectada de varios proyectos.

La síntesis también fue aplicada en la generación de la gramática del DSL, una vez realizado el análisis de la aplicación modelo.

Para la evaluación de la productividad también fue necesario el uso del método sintético.

### 3.1.1.3 *Técnicas de investigación*

Las técnicas que se utilizaron en este trabajo de integración curricular son las que se detallan a continuación:

- Para el análisis de las de las herramientas se utilizó la revisión de documentos y entrevistas con profesionales
- Para el análisis de sistemas realizados con el lenguaje Python es decir el estudio del dominio de la aplicación se utilizó la observación
- Para el desarrollo de la aplicación modelo se utilizó la revisión de documentos
- Para desarrollar la gramática del lenguaje se utilizó la revisión de documentos y entrevistas con profesionales.
- Para evaluar la productividad se utilizó la revisión de documentos y técnicas estadísticas para evaluar los resultados, como métrica se tomó en cuenta la metodología Personal Software Process la cual propone el número de líneas de código por hora (LOCs/hr) como métrica para evaluar la productividad.
- Para evaluar los costos de desarrollo se utilizó la revisión de documentos y técnicas estadísticas para evaluar los resultados, el valor tomado en cuenta para la evaluación fue el establecido por el Ministerio de Trabajo del Ecuador para los salarios básicos de los desarrolladores junior en el 2021.

## 3.2 **Metodología para el desarrollo del DSL**

La metodología para la gestión del desarrollo que se usó fue *SCRUM* la cual permitió realizar la planificación de actividades, llevar a cabo las iteraciones, presentar avances del DSL de tal forma que se pueda identificar a tiempo cualquier percance en el desarrollo y corregirlo de ser necesario

La metodología de desarrollo que se usó para el DSL fue la propuesta por Gómez et al. en su artículo “CRUDyLeaf: A DSL for Generating Spring Boot REST APIs from Entity CRUD Operations”(Gómez et al., 2020) la cual a través de sus etapas nos permitió desarrollar el DSL sin tener percances ya que se detallan claramente los módulos que se deben realizar al crear un DSL, para la evaluación del DSL se desarrolló dos casos de prueba, uno con conexión a una base de datos relacional y otro con conexión a una base de datos no relacional.

Las etapas de la metodología que se usó para el desarrollo del DSL son: estudio del dominio de aplicación, implementación de la aplicación modelo, definición de la gramática del lenguaje, generación del metamodelo y la implementación del generador de código. En el estudio de dominio se realizó el análisis de proyectos desarrollados con determinadas tecnologías, una vez definida la estructura con la cual se desarrollan los proyectos se procedió a la implementación de la aplicación modelo, con la cual a través del estudio de sus elementos se definió la gramática que usara el DSL.

### **3.3 Fase de planificación**

En esta fase se estimó el tiempo necesario para realizar las actividades en el desarrollo del DSL, estas actividades son: revisión de literatura, documentación, análisis preliminar, planificación, desarrollo y la evaluación del DSL, a su vez estas actividades se subdividen para detallar algunos procesos que se desarrollaron en dichas actividades.

Esta planificación ha sido representada en un diagrama de Gantt en el cual se muestran las actividades con sus respectivas fechas estimadas para la realización, esta planificación se muestra en la figura 3-1.



**Figura 3-1:** Diagrama de Gantt

**Fuente:** Núñez, Gilberth, 2021

### 3.3.1 Personas y roles del proyecto

El equipo que se encargó del desarrollo del proyecto está conformado por dos personas las cuales sus roles correspondientes se muestran en la siguiente tabla.

**Tabla 1-3:** Roles y personas

Persona	Contacto	Rol
Dr. Omar Salvador Gómez Gómez	<a href="mailto:ogomez@epoch.edu.ec">ogomez@epoch.edu.ec</a>	Scrum Master
Gilberth Ianrolando Núñez Ribadeneyra	<a href="mailto:gilberth.nunez@epoch.edu.ec">gilberth.nunez@epoch.edu.ec</a>	Team Development

**Realizado por:** Núñez, Gilberth, 2021

### 3.3.2 Product backlog

A continuación, se muestra una lista de características las cuales tienen su nivel de prioridad y contienen breves descripciones, esto nos permite estimar e identificar que funcionalidades deben ser priorizadas, estas características serán descritas a través de historias técnicas.

Los puntos de esfuerzo fueron determinados a través de la técnica *Planning Poker*, esto nos permitirá aproximar los tiempos de desarrollo. En la siguiente tabla se muestra los valores que fueron tomados en cuenta para el desarrollo de esta estimación.

**Tabla 2-3: Planning poker**

<b>Planning poker</b>		
1/8 iteraciones	1	1 punto
1/4 iteraciones	2	2 puntos
1/2 iteraciones	3	3 puntos
1 iteraciones	5	5 puntos
2 iteraciones	8	8 puntos
+ de 2 iteraciones	13	13 puntos

**Realizado por:** Núñez, Gilberth, 2021

Cada punto estimado tiene su equivalencia respecto a la duración de las iteraciones, teniendo en cuenta que el tiempo de cada iteración consta de 4 semanas, se trabajó los días laborables es decir de lunes a viernes 4 horas diarias.

A continuación, se muestra en la tabla 3-3 de las historias técnicas identificadas para el desarrollo del DSL.

**Tabla 3-3: Product backlog**

<b>ID</b>	<b>Descripción</b>	<b>Puntos de Esfuerzo (pts)</b>
<b>HT_01</b>	Como desarrollador se desea recolectar información	2 pts
<b>HT_02</b>	Como desarrollador se desea seleccionar e instalar las herramientas para el desarrollo del DSL	3 pts
<b>HT_03</b>	Como desarrollador se desea identificar la estructura general de proyectos desarrollados en el lenguaje Python	2 pts
<b>HT_04</b>	Como desarrollador se desea diseñar la base de datos	1 pts
<b>HT_05</b>	Como desarrollador se desea definir el estándar de programación	1 pts
<b>HT_06</b>	Como desarrollador se desea estudiar el dominio de la aplicación	8 pts
<b>HT_07</b>	Como desarrollador se desea desarrollar la aplicación modelo	13 pts
<b>HT_08</b>	Como desarrollador se desea desarrollar la gramática del lenguaje	5 pts
<b>HT_09</b>	Como desarrollador se desea generar el metamodelo	1 pts
<b>HT_10</b>	Como desarrollador se desea desarrollar el generador de código	8 pts
<b>HT_11</b>	Como desarrollador se desea generar un proyecto con el DSL	1 pts

**Realizado por:** Núñez, Gilberth, 2021

En total se obtuvo 11 historias técnicas las mismas que se distribuirán en los *sprints*.

### 3.3.3 *Sprint backlog*

En esta etapa se asignó tareas del *product backlog* a un determinado *sprint*, es decir que una o varias historias técnicas serán asignadas a un determinado *sprint*.

El desarrollo del DSL consta de 8 *sprints*, cada *sprint* tuvo la duración de 4 semanas, es decir un total de 5 puntos de esfuerzo estimados.

**Tabla 4-3:** Sprint backlog

N° Sprint	ID	Requerimientos	Inicio	Fin	Puntos Estimados (pts)	Puntos Totales (pts)
1	HT_01	Como desarrollador se desea recolectar información	15/06/2020	19/06/2020	2 pts	7 pts
	HT_02	Como desarrollador se desea seleccionar e instalar las herramientas para el desarrollo del DSL	22/06/2020	03/07/2020	3 pts	
	HT_03	Como desarrollador se desea identificar la estructura general de proyectos desarrollados en el lenguaje Python	06/07/2020	10/07/2020	2 pts	
2	HT_04	Como desarrollador se desea diseñar la base de datos	13/07/2020	11/09/2020	1 pts	10 pts
	HT_05	Como desarrollador se desea definir el estándar de programación	16/07/2020	11/09/2020	1pts	
	HT_06	Como desarrollador se desea estudiar el dominio de la aplicación	20/07/2020	11/09/2020	8 pts	
3						5 pts
4						7 pts
5	HT_07	Como desarrollador se desea desarrollar la aplicación modelo	14/09/2020	06/11/2020	13 pts	5 pts
6						10 pts
7	HT_08	Como desarrollador se desea desarrollar la gramática del lenguaje	09/11/2020	04/12/2020	5 pts	11 pts
	HT_09	Como desarrollador se desea generar el metamodelo	07/12/2020	08/12/2020	1 pts	
	HT_10	Como desarrollador se desea desarrollar el generador de código	09/12/2020	19/01/2021	8 pts	
8						6 pts
	HT_11	Como desarrollador se desea generar un proyecto con el DSL	20/01/2021	22/01/2021	1 pts	

Realizado por: Núñez, Gilberth, 2021

### 3.3.4 Planificación de la gestión de riesgos

En esta etapa se contempló los posibles riesgos que se pueden presentar en el desarrollo del proyecto, de esta forma poder gestionarlos en una etapa temprana para minimizarlos, controlarlos y evadirlos.

Primero se identificó cuáles serán los valores de impacto que nos permitieron determinar la prioridad con la cual se tratará cada uno de los riesgos.

**Tabla 5-3:** Valores de impacto

<b>Equivalencia</b>	<b>Valor</b>	<b>Descripción</b>
1	Catastrófico	Perdida del sistema. Costo >50%
2	Critico	Recuperación de la capacidad operativa. Costo >15% (<50%)
3	Marginal	Costo >5% (<15%)
4	Despreciable	Costo <5%

**Realizado por:** Núñez, Gilberth, 2021

Una vez establecidos los valores con los cuales se dará prioridad a los riesgos se procedió a evaluar los riesgos, el indicador representará la presencia de un riesgo y las consecuencias de este. Según su prioridad se desarrollará una estrategia para tratar este riesgo, en la tabla 6-3 se muestran los riesgos del proyecto.

**Tabla 6-3:** Tabla de riesgos

<b>Riesgos</b>	<b>Categoría</b>	<b>Probabilidad</b>	<b>Impacto</b>
La estimación del tamaño puede ser significativamente baja	Tamaño del producto	20%	2
Carencia en el uso de las herramientas	Entorno de desarrollo	30%	3
La tecnología no alcanzara las expectativas	Tecnología por construir	10%	1
Diseño inadecuado	Proceso de producción	30%	3

**Realizado por:** Núñez, Gilberth, 2021

Una vez identificados los riesgos del proyecto se procedió a establecer estrategias para minimizar la probabilidad de que este suceda. Las estrategias por riesgo planteadas se muestran en la tabla 7-3.



**Tabla 7-3:** Estrategias por riesgos

<b>Riesgos</b>	<b>Estrategia</b>
La estimación del tamaño puede ser significativamente baja	Analizar los procesos de desarrollo del DSL y agregar un margen de error a la estimación.
Carencia en el uso de las herramientas	Indagar previamente sobre el uso de las herramientas, para de esta forma sacar el mayor provecho.
La tecnología no alcanzara las expectativas	Analizar si la tecnología que fue seleccionada para el desarrollo del proyecto es la adecuada y se han desarrollado proyectos similares
Diseño inadecuado	Analizar la estructura de proyectos similares, la cual será implementada a los proyectos generados por el DSL

Realizado por: Núñez, Gilberth, 2021

### 3.3.5 Recursos necesarios

Para el desarrollo del DSL se utilizó recursos hardware, software, materiales y otros recursos los cuales se detallan en las tablas a continuación

Como recurso hardware se utilizó una laptop para desarrollar funciones como programar, documentar y realizar pruebas, tal como se muestra en la siguiente tabla.

**Tabla 8-3:** Recurso hardware

<b>Cantidad</b>	<b>Descripción</b>	<b>Función</b>	<b>Áreas</b>
1	Computadora DELL Windows 10 Home, procesador Intel(R) Core (TM) i5-4550U CPU @ 1.80GHz 1.99GHz, memoria RAM 4,00 GB, almacenamiento 250GB	Programación, documentación y pruebas	Análisis Mantenimiento

Realizado por: Núñez, Gilberth, 2021

Para el desarrollo del DSL como recurso software se utilizó una serie de recursos los cuales se muestran en la tabla 9-3.

**Tabla 9-3:** Recurso software

<b>IDE de desarrollo</b>	Eclipse 2019-12
<b>Sistema Operativo</b>	Sistema Operativo Windows 10 Home
<b>Sistema Gestor de Base de Datos</b>	Postgresql 9.4 MongoDB
<b>Navegador</b>	Mozilla Firefox
<b>Software Ofimática</b>	Microsoft Office 2016
<b>Framework</b>	Flask, Xtext
<b>Lenguaje de programación</b>	Python, Xtend

Realizado por: Núñez, Gilberth, 2021

### 3.4 Fase de Desarrollo, generación de la aplicación modelo y definición de la gramática

#### 3.4.1 Estándar de codificación

Con el objetivo de que el código sea uniforme se definió un estándar el cual facilitara la legibilidad y el mantenimiento de este para ello se propuso lo mostrado en la tabla 10-3

**Tabla 10-3:** Estándar de codificación

Componente	Archivo	Nombre	Descripción
Clases	Models	<NOMBRECLASE>  < PERSONA >	El nombre de la clase debe hacer referencia a que trata, cada letra de la clase debe ir en mayúscula
	Serializers	<NOMBRECLASESchema>  < PERSONASchema>	El nombre de la clase debe hacer referencia a que trata, cada letra de la clase debe ir en mayúscula y siempre al final se escribirá la palabra Schema
	idResource	<idnOMBRECLASE>  < idpPERSONA>	El nombre de la clase debe hacer referencia a que trata, se debe anteponer las letras “id” en minúscula a continuación la primera letra de la clase debe ir en minúscula y seguido cada letra de la clase debe ir en mayúscula.
	Resource	<nOMBRECLASE>  < pPERSONA>	El nombre de la clase debe hacer referencia a que trata, la primera letra de la clase debe ir en minúscula y seguido cada letra de la clase debe ir en mayúscula.
Variables	Todos	<nombrevariable>  < detalle >	La variable debe ser escrita en minúscula cada una de sus letras
Métodos	Todos	<nombremetodo>  < post >	La variable debe ser escrita en minúscula cada una de sus letras

Realizado por: Núñez, Gilberth, 2021

### 3.4.2 Desarrollo de los sprints

**Sprint N° 1:** Mediante reuniones con el *scrum* máster se pudo recolectar información sobre las herramientas y tecnología necesaria para desarrollar el sistema, además se identificó la estructura general de proyectos desarrollados en *python*.

**Tabla 11-3:** Sprint 1

<b>Sprint 1</b>					
<b>Inicio:</b> 15/06/2020		<b>Fin:</b> 10/07/2020		<b>Esfuerzo Estimado:</b> 7 pts	<b>Esfuerzo Real:</b> 7 pts
<b>Pila del Sprint</b>					
<b>Product backlog ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>	
HT_01	Como desarrollador se desea recolectar información	2 pts	Análisis	Gilberth Núñez	
HT_02	Como desarrollador se desea seleccionar e instalar las herramientas para el desarrollo del DSL	3 pts	Análisis	Gilberth Núñez	
HT_03	Como desarrollador se desea identificar la estructura general de proyectos desarrollados en el lenguaje Python	2 pts	Análisis	Gilberth Núñez	

**Realizado por:** Núñez, Gilberth, 2021

**Sprint N° 2:** En este *sprint* se realizó el diseño de la base de datos, el estándar de codificación y se procedió con el estudio del dominio de la aplicación, específicamente como iniciar un proyecto en el lenguaje Python con el uso del micro-framework flask.

**Tabla 12-3: Sprint 2**

<b>Sprint 2</b>					
<b>Inicio:</b> 13/07/2020		<b>Fin:</b> 07/08/2020		<b>Esfuerzo Estimado:</b> 7 pts	<b>Esfuerzo Real:</b> 7 pts
<b>Pila del Sprint</b>					
<b>Product backlog ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>	
HT_04	Como desarrollador se desea diseñar la base de datos	1 pts	Análisis Diseño	Gilberth Núñez	
HT_05	Como desarrollador se desea definir el estándar de programación	1 pts	Análisis	Gilberth Núñez	
HT_06	Como desarrollador se desea estudiar el dominio de la aplicación	5 pts	Análisis Codificación	Gilberth Núñez	

**Realizado por:** Núñez, Gilberth, 2021

**Sprint N° 3:** En este sprint se continuó con el estudio del dominio de la aplicación en la creación del archivo modelo y serializadores.

**Tabla 13-3: Sprint 3**

<b>Sprint 3</b>					
<b>Inicio:</b> 10/08/2020		<b>Fin:</b> 04/09/2020		<b>Esfuerzo Estimado:</b> 5 pts	<b>Esfuerzo Real:</b> 5 pts
<b>Pila del Sprint</b>					
<b>Product backlog ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>	
HT_06	Como desarrollador se desea estudiar el dominio de la aplicación	5 pts	Análisis Codificación	Gilberth Núñez	

**Realizado por:** Núñez, Gilberth, 2021

**Sprint N° 4:** En este *sprint* finalizó el estudio de dominio de la aplicación la declaración y creación de servicios REST y se procedió a desarrollar la aplicación modelo creando el archivo main y de configuración.

**Tabla 14-3:** Sprint 4

Sprint 4					
<b>Inicio:</b> 07/09/2020		<b>Fin:</b> 02/10/2020		<b>Esfuerzo Estimado:</b> 7 pts	<b>Esfuerzo Real:</b> 7 pts
Pila del Sprint					
Product backlog ID	Descripción	Esfuerzo	Tipo	Responsable	
HT_06	Como desarrollador se desea estudiar el dominio de la aplicación	2 pts	Análisis Codificación	Gilberth Núñez	
HT_07	Como desarrollador se desea desarrollar la aplicación modelo	5 pts	Codificación	Gilberth Núñez	

**Realizado por:** Núñez, Gilberth, 2021

**Sprint N° 5:** En este *sprint* se continuó con el desarrollo de la aplicación modelo en la creación del archivo modelo.

**Tabla 15-3:** Sprint 5

Sprint 5					
<b>Inicio:</b> 05/10/2020		<b>Fin:</b> 30/10/2020		<b>Esfuerzo Estimado:</b> 5 pts	<b>Esfuerzo Real:</b> 5 pts
Pila del Sprint					
Product backlog ID	Descripción	Esfuerzo	Tipo	Responsable	
HT_07	Como desarrollador se desea desarrollar la aplicación modelo	5 pts	Codificación	Gilberth Núñez	

**Realizado por:** Núñez, Gilberth, 2021

**Sprint N° 6:** En este *sprint* se continuó con el desarrollo de la aplicación modelo en la creación del archivo serializadores

**Tabla 16-3: Sprint 6**

<b>Sprint 6</b>					
<b>Inicio:</b> 02/11/2020		<b>Fin:</b> 27/11/2020		<b>Esfuerzo Estimado:</b> 10 pts	<b>Esfuerzo Real:</b> 10 pts
<b>Pila del Sprint</b>					
<b>Product backlog ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>	
HT_07	Como desarrollador se desea desarrollar la aplicación modelo	5 pts	Codificación	Gilberth Núñez	
HT_08	Como desarrollador se desea desarrollar la gramática del lenguaje	5 pts	Codificación	Gilberth Núñez	

**Realizado por:** Núñez, Gilberth, 2021

**Sprint N° 7:** En este *sprint* se continuó con el desarrollo de la aplicación modelo en la creación y declaración de servicios REST, se procedió con la generación del metamodelo y se procedió a desarrollar el generador de código para proyectos con conexión a una base de datos relacional

**Tabla 17-3: Sprint 7**

<b>Sprint 7</b>					
<b>Inicio:</b> 30/11/2020		<b>Fin:</b> 25/12/2020		<b>Esfuerzo Estimado:</b> 11 pts	<b>Esfuerzo Real:</b> 11 pts
<b>Pila del Sprint</b>					
<b>Product backlog ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>	
HT_08	Como desarrollador se desea desarrollar la gramática del lenguaje	5 pts	Codificación	Gilberth Núñez	
HT_09	Como desarrollador se desea generar el metamodelo	1 pts	Diseño	Gilberth Núñez	
HT_10	Como desarrollador se desea desarrollar el generador de código	5 pts	Diseño	Gilberth Núñez	

**Realizado por:** Núñez, Gilberth, 2021

**Sprint N° 8:** En este *sprint* finalizó el proceso del desarrollo de la aplicación modelo para proyectos con conexión a una base de datos no relacional y finalmente se generó un proyecto con el uso del DSL.

**Tabla 18-3: Sprint 8**

<b>Sprint 8</b>				
<b>Inicio:</b> 28/12/2020	<b>Fin:</b> 22/01/2020	<b>Esfuerzo Estimado:</b> 6 pts	<b>Esfuerzo Real:</b> 6 pts	
<b>Pila del Sprint</b>				
<b>Product backlog ID</b>	<b>Descripción</b>	<b>Esfuerzo</b>	<b>Tipo</b>	<b>Responsable</b>
HT_10	Como desarrollador se desea desarrollar el generador de código	5 pts	Diseño	Gilberth Núñez
HT_11	Como desarrollador se desea generar un proyecto con el DSL	1 pts	Codificación	Gilberth Núñez

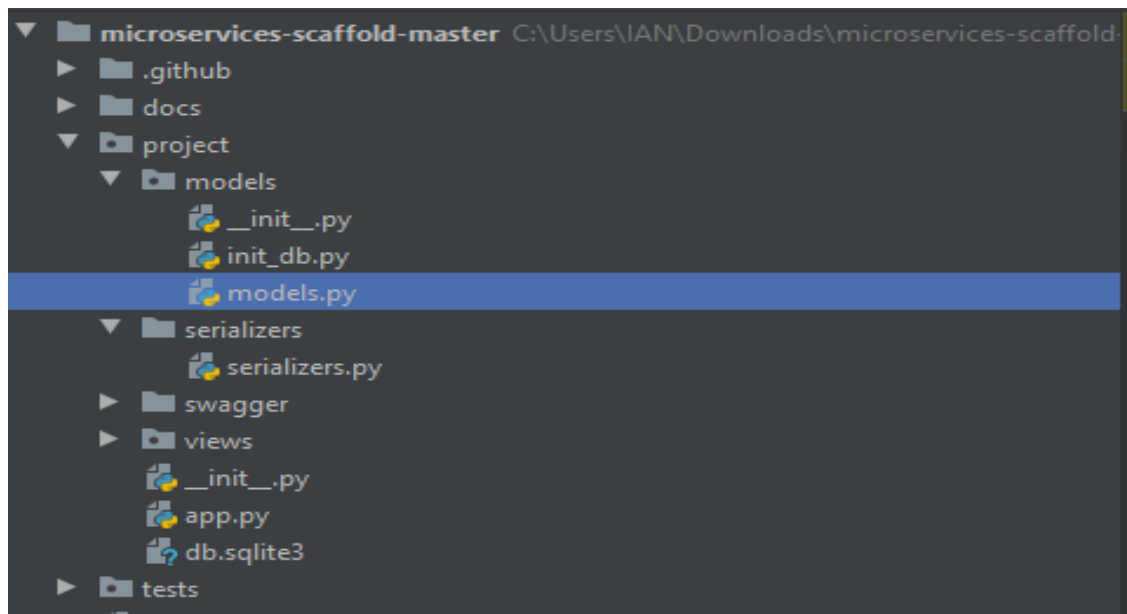
**Realizado por:** Núñez, Gilberth, 2021

### **3.4.3 Estudio del dominio de aplicación del DSL**

Una vez establecida la planificación de las actividades a realizar pasamos a el estudio de dominio la cual es definida por Gómez et al. como:

*Esta etapa consiste en conocer el dominio en el que se construirá el DSL. En nuestro caso, examinamos el marco de trabajo de Spring Boot, en particular cómo se implementan las típicas aplicaciones REST con operaciones CRUD. Se codificaron una serie de ejercicios para comprender mejor este dominio.(Gómez et al., 2020, p. 6)*

En esta etapa se indagó sobre cómo se implementan las aplicaciones escritas en el lenguaje Python con el estilo de arquitectura REST y operaciones CRUD. De los proyectos encontrados se identificó una estructura en común mostrada en la figura 2-3, esta estructura se tomó en cuenta para formar la aplicación modelo.



**Figura 3-2:** Estructura de un proyecto desarrollado en el lenguaje Python

Fuente: Núñez, Gilberth, 2021

En la estructura se identificó un archivo que ejecuta la configuración con el servidor de base de datos y la aplicación, los proyectos que trabajan con un ORM trabajan con una carpeta denominada *models* y *serializers*, en la carpeta *models* se encontró el archivo “.py” en el cual se declaran las clases que serán comunicadas a la base de datos y crear sus respectivas tablas, en la carpeta *serializers* se encontró el archivo “.py” en el cual se define la estructura de los datos que serán serializados.

#### 3.4.4 Implementación de la aplicación modelo (gold application)

Una vez realizado el estudio de dominio e identificado la estructura y las herramientas utilizadas comúnmente en el desarrollo de proyectos en el lenguaje *Python* con el *microframework Flask* realizamos la aplicación modelo de la aplicación, esta etapa es descrita por Gómez et al como:

*Una vez inmersos en el dominio de interés, implementamos una aplicación Spring Boot que expone las operaciones CRUD como recursos REST. Esta aplicación se utilizó como plantilla dorada para derivar la sintaxis del DSL, esta aplicación dorada también se utiliza como referencia para implementar el generador de código.*(Gómez et al., 2020, p. 6)

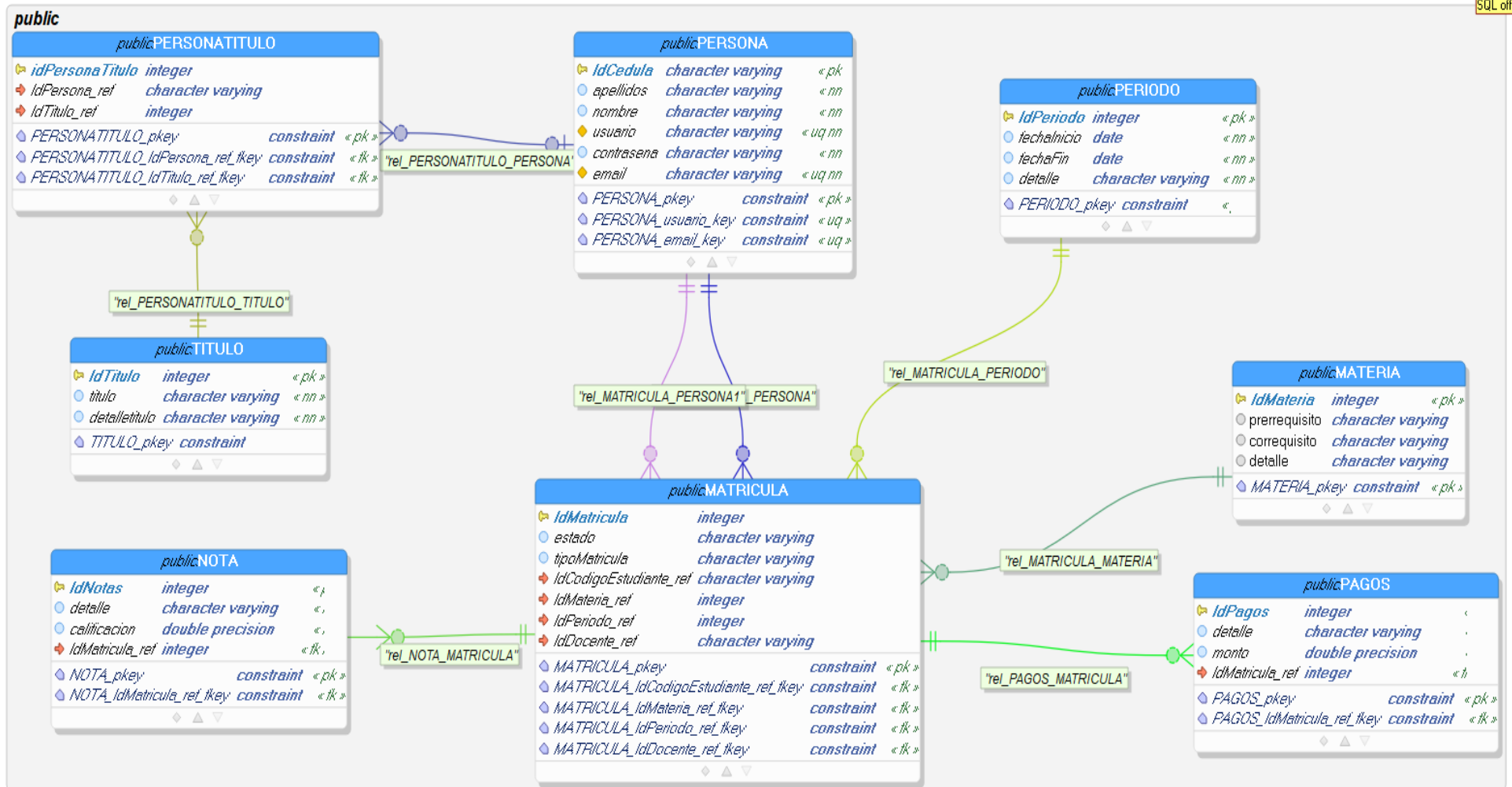
Se elaboró una aplicación modelo partiendo del diseño de la base de datos de un sistema académico tomado como ejemplo, se diseñó el modelo Entidad-Relación el cual nos permitió realizar el modelado de datos, luego se realizó la normalización de la base de datos que nos



permite minimizar la redundancia de datos, disminuir problemas de actualización de datos y proteger la integridad de datos, de esta forma se obtuvo el modelo lógico.

En el diagrama lógico constó de 8 tablas con sus respectivas relaciones, claves primarias, claves foráneas y campos, los nombres de dichas tablas están escritas en mayúsculas Ej. PERSONA, el nombre de los campos en minúsculas Ej. nombre, para identificar una clave primaria se utilizó el acrónimo “Id” en referencia a identificador y a continuación el nombre del campo la primera letra de cada palabra en mayúscula Ej. IdCedula, para identificar una clave foránea se utilizó el acrónimo “Id” a continuación el nombre del campo con la inicial de cada palabra en mayúscula y debe terminar con un “\_ref” Ej. IdTitulo\_ref.

Resultado del anterior análisis se diseñó el modelo lógico de la base de datos el cual se muestra en la figura 2-4



**Figura 3-3:** Modelo lógico de la base de datos

Fuente: Núñez, Gilberth, 2021

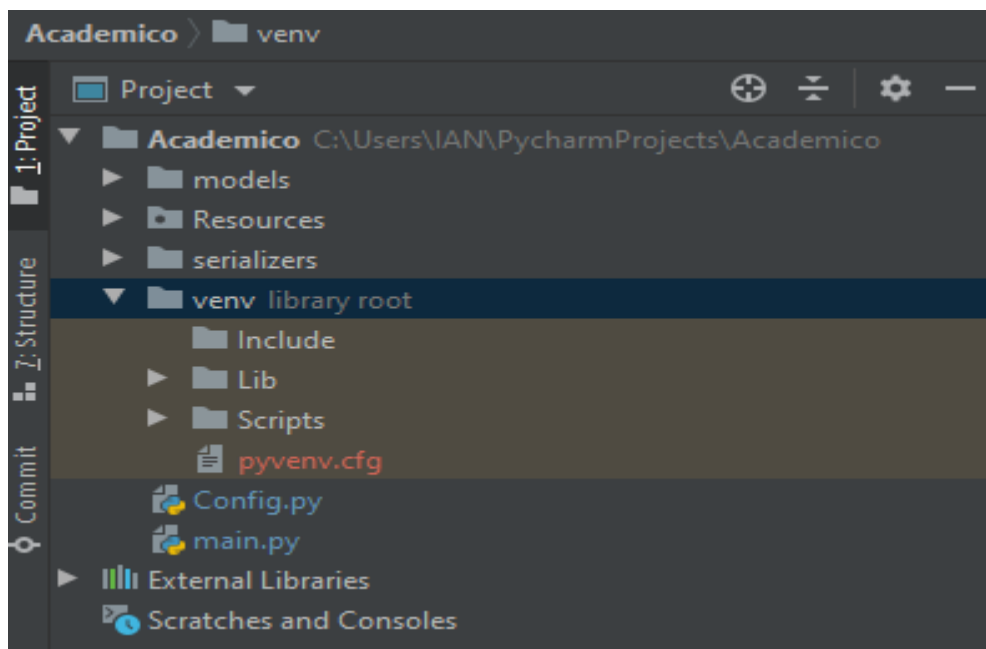
**Tabla 19-3:** Diccionario de datos de la tabla persona

public.PERSONA							Table
Name	Data type	PK	FK	UQ	Not null	Default value	Description
IdCedula	character varying	✓			✓		
apellidos	character varying				✓		
nombre	character varying				✓		
usuario	character varying			✓	✓		
contrasena	character varying				✓		
email	character varying			✓	✓		
Constraints							
Name	Type	Column(s)	References	Expression	Description		
PERSONA_pkey	PRIMARY KEY	IdCedula					
PERSONA_usuario_key	UNIQUE	usuario					
PERSONA_email_key	UNIQUE	email					

Fuente: Núñez, Gilberth, 2021

La totalidad del diccionario de datos se encuentra en el Anexo A.

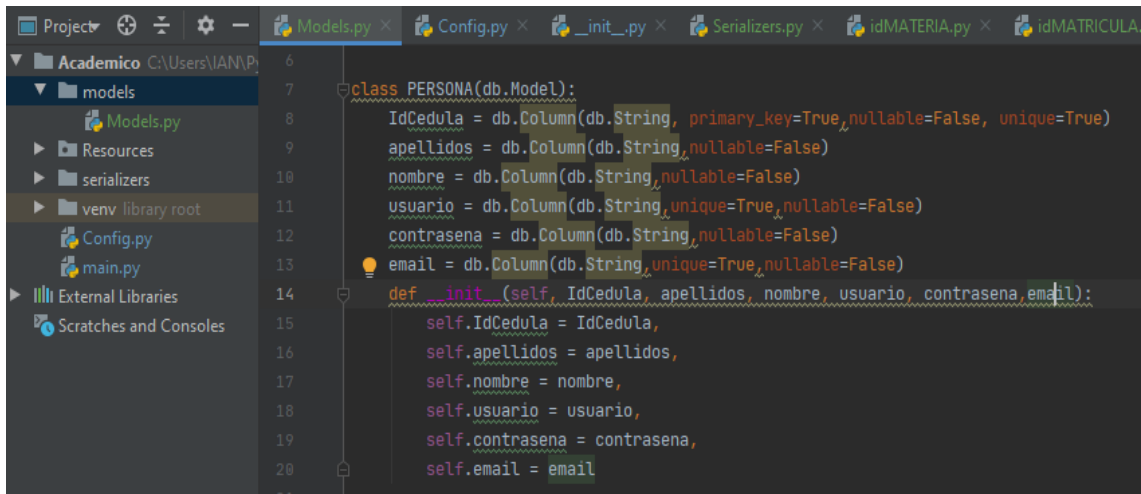
Una vez definido el diseño de la base de datos, se procedió a crear la estructura de la aplicación modelo en el *IDE Pycharm* con su respectivo entorno virtual en el cual se instalaron los paquetes *Python* necesarios para el funcionamiento del sistema, dicha estructura se muestra en la siguiente figura.



**Figura 3-4:** Estructura de un proyecto modelo

Fuente: Núñez, Gilberth, 2021

En la carpeta *models* se encuentra el archivo “.py” que contiene la declaración de las clases las cuales serán comunicadas a la base de datos para su creación o modificación

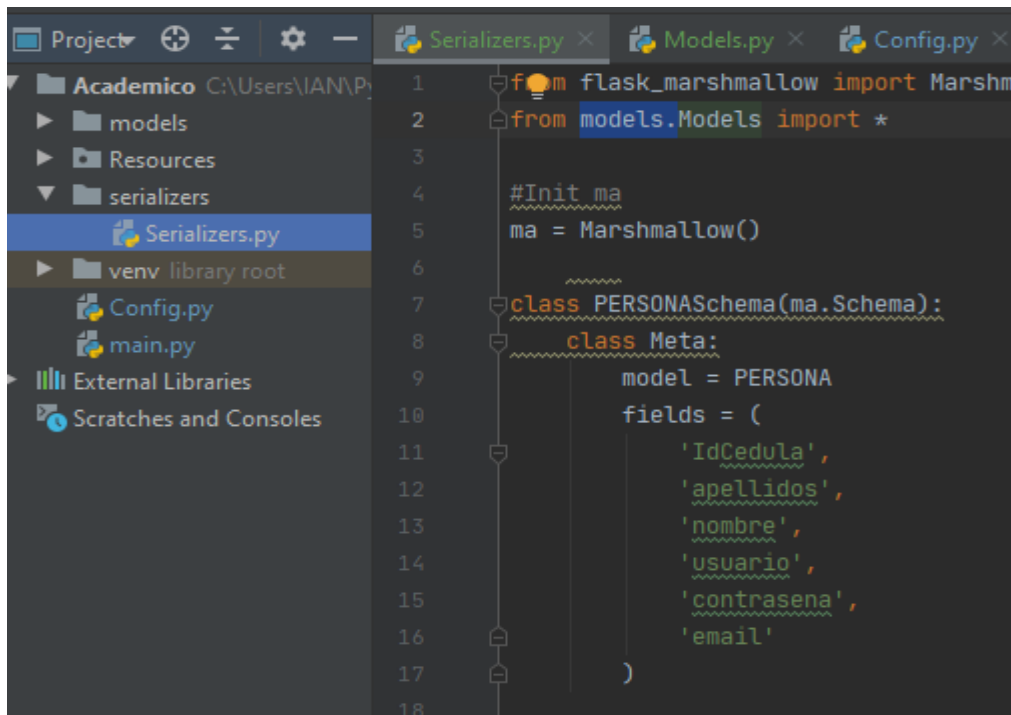


```
6
7 class PERSONA(db.Model):
8     IdCedula = db.Column(db.String, primary_key=True, nullable=False, unique=True)
9     apellidos = db.Column(db.String, nullable=False)
10    nombre = db.Column(db.String, nullable=False)
11    usuario = db.Column(db.String, unique=True, nullable=False)
12    contrasena = db.Column(db.String, nullable=False)
13    email = db.Column(db.String, unique=True, nullable=False)
14
15    def __init__(self, IdCedula, apellidos, nombre, usuario, contrasena, email):
16        self.IdCedula = IdCedula,
17        self.apellidos = apellidos,
18        self.nombre = nombre,
19        self.usuario = usuario,
20        self.contrasena = contrasena,
21        self.email = email
```

**Figura 3-5:** Extracto del código de creación de clases

Fuente: Núñez, Gilberth, 2021

En la carpeta *serializers* se encuentra el archivo “.py” que contiene la declaración de la estructura de los datos que serán serializados



```
1 from flask_marshmallow import Marshmallow
2 from models.Models import *
3
4 #Init ma
5 ma = Marshmallow()
6
7 class PERSONASchema(ma.Schema):
8     class Meta:
9         model = PERSONA
10        fields = (
11            'IdCedula',
12            'apellidos',
13            'nombre',
14            'usuario',
15            'contrasena',
16            'email'
17        )
18
```

**Figura 3-6:** Extracto del código de la declaración de estructura de datos

Fuente: Núñez, Gilberth, 2021

En la carpeta *Resources* se encuentran los archivos “.py” en el cual se define la ruta a la cual se debe hacer referencia para consumir los recursos, así como también se encuentran los recursos con cada uno de sus métodos.

Se generó dos archivos “.py” por clase declarada, el nombre generado para estos recursos son el nombre de la clase en mayúsculas que contiene la declaración del recurso que trabaja con los métodos sin un parámetro, estos métodos son el POST y el GET, el segundo archivo generado posee el mismo nombre que el anterior anteponiendo las letras “id” en minúsculas, este archivo contiene la declaración del recurso que trabaja con un parámetro en sus métodos, estos métodos son GET, PUT y DELETE.

El archivo que tiene la declaración de las rutas hacia estos recursos tiene el nombre de “\_\_init\_\_.py” en el cual se especifica la ruta con la cual el cliente accede a los recursos ahí declarados.

En caso de un proyecto con conexión a una base de datos tipo “Mongo” se generó un archivo adicional el cual posee el método que permite enviar el id generado por mongo hacia el cliente.

### **3.4.5 Definición de la gramática del lenguaje**

Una vez desarrollada la aplicación modelo y definido las herramientas que interactúan en un sistema que trabaja con un ORM y recursos RESTful, se procede a la definición de la gramática del lenguaje, en esta etapa las actividades a desarrollar son descritas por Gómez et al. como:

*Con una versión primitiva de la sintaxis, en esta etapa refinamos y terminamos la sintaxis concreta y construimos el DSL. Lo implementamos utilizando Xtext. Xtext es un potente marco para construir bancos de trabajo de lenguaje para DSL textuales. Xtext sólo requiere especificar un archivo de gramática. Con este archivo de sintaxis, Xtext define el lenguaje y crea la infraestructura necesaria, como el analizador sintáctico, el enlazador, el comprobador de tipos, así como incluye un editor para el IDE de Eclipse. (Gómez et al., 2020, p. 6)*

Se elaboró la gramática para trabajar con proyectos conectados a base de datos de tipo relacional como “PostgreSQL” y no relacional como “MongoDB” una vez seleccionado el tipo de base de datos, se debe ingresar las credenciales que permiten la conexión a esta, los datos solicitados son el nombre de usuario, la contraseña, la dirección, el puerto y el tipo de base de datos.

```

38 DB:
39     MongoDB | PostgreSQL;
40
41
42 PostgreSQL:
43     'DBMS' name = 'postgresql' '{'
44         'username' '=' username = UserName
45         'password' '=' password = Password
46         'host' '=' host = Host
47         'port' '=' port = Port
48         'database' '=' database = Database
49     '}'
50
51     class += ClassPostgreSQL*;
52
53
54 MongoDB:
55     'DBMS' name = 'mongodb' '{'
56         'username' '=' username = UserName
57         'password' '=' password = Password
58         'host' '=' host = Host
59         'port' '=' port = Port
60         'database' '=' database = Database
61     '}'
62
63     class += ClassMongoDB*
64 ;

```

**Figura 3-7:** Extracto del código de la gramática del DSL

Fuente: Núñez, Gilberth, 2021

Una vez seleccionado el tipo de base de datos se procede a declarar las clases, para definir una clase se debe utilizar la palabra reservada “*class*” después el nombre de la clase, estas clases poseen atributos y estos atributos a su vez poseen características como el tipo de dato, si será un atributo auto incrementable, si este permite valores nulos o no, estas características poseen un valor el cual se establece como verdadero o falso al momento de su declaración.

Tanto para el uso de una base de datos de tipo “PostgreSQL” como “MongoDB” tienen la misma estructura respecto a los datos solicitados para la conexión, pero poseen una gramática distinta entre los dos respecto a sus atributos y las características pertenecientes a estos atributos, la información más a detalle sobre la gramática se encuentra en el Anexo E.

### 3.4.6 Generación del metamodelo

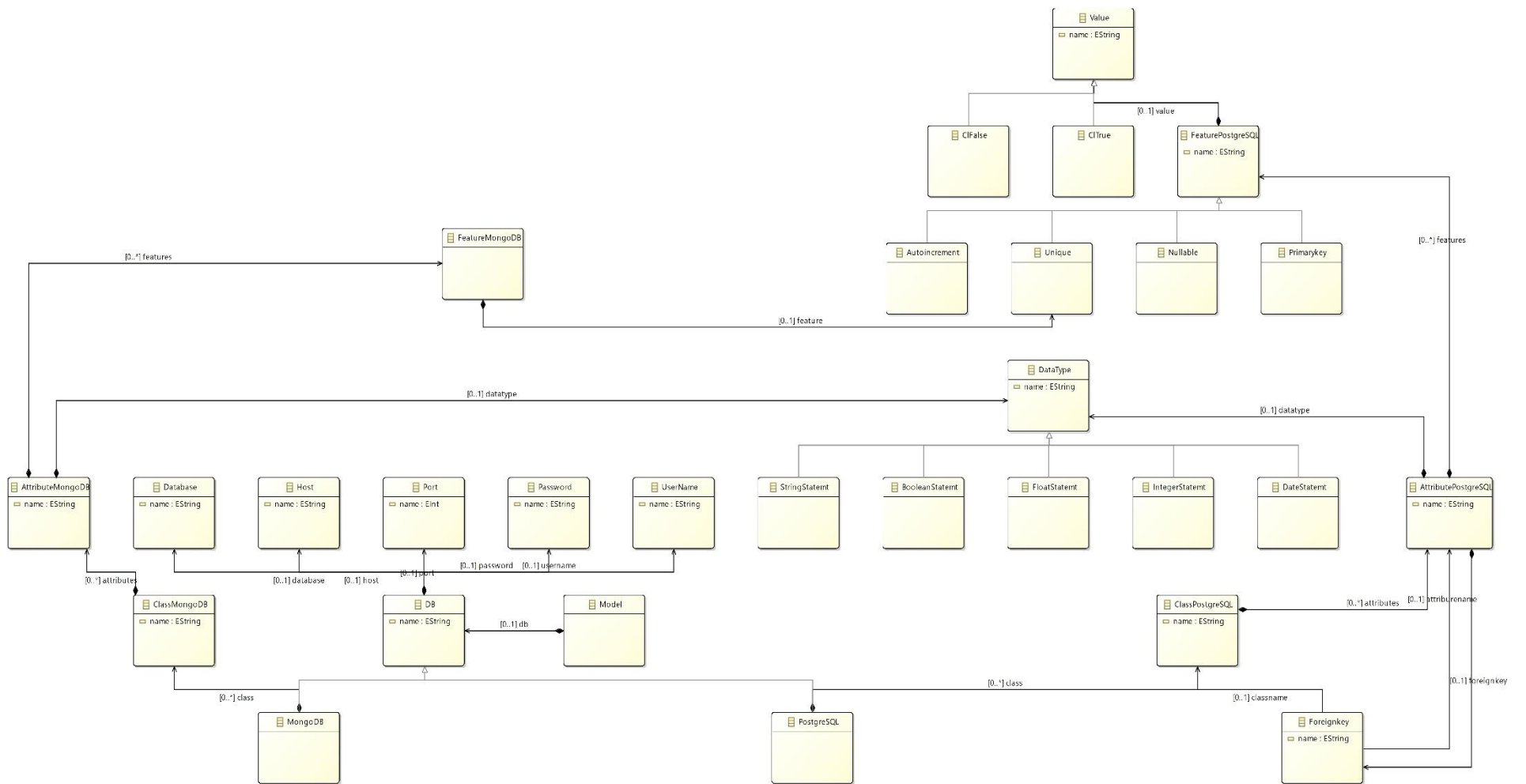
Después de realizar la definición de la gramática del lenguaje procedemos a la generación del metamodelo, esta etapa es descrita por Gómez et al. como:

*La sintaxis concreta sirve de entrada para generar el modelo semántico (metamodelo). Entre la infraestructura generada por Xtext, se genera un modelo Ecore.*

*Representa el árbol sintáctico abstracto (AST). Este modelo Ecore puede utilizarse para realizar transformaciones de Modelo a Texto (M2T), es decir, generar código fuente a partir del modelo.*

*Esta etapa se realiza automáticamente en el IDE de Eclipse ejecutando la función de generación de artefactos Xtext.* (Gómez et al., 2020, pp. 6–7)

Para la generación del modelo *Ecore* se utilizó la función de generación de artefactos Xtext en el IDE Eclipse. Este modelo nos permite comunicar de una mejor forma la estructura de nuestra gramática



**Figura 3-8:** Ecore metamodelo extraído del DSL

Fuente: Núñez, Gilberth, 2021



### 3.4.7 Implementación del generador de código

Una vez generado el metamodelo se procedió a la implementación del generador de código, esta etapa es descrita por Gómez et al. como “Hemos utilizado el lenguaje Xtend para escribir el generador de código del DSL propuesto. Xtend es un lenguaje de programación de tipado estático que se publicó inicialmente con Xtext“(Gómez et al., 2020, p. 8)

El generador de código permitió obtener como salida el proyecto CRUD-REST el cual usa como archivo de entrada la declaración de un proyecto con la gramática propuesta.

En la figura 3-9 se muestra un extracto del lenguaje Xtend para la generación del método PUT en la declaración de un servicio.

```
682 def putMongo(ClassMongoDB c) '''
683     «"\t"»# Update «c.name»
684     «"\t"»def put(self, id«c.name.toUpperCase»):
685         «"\t\t"»«c.name.toLowerCase»_query = «c.name.toUpperCase».query.get(id«c.name.toUpperCase»)
686
687     «FOR a : c.attributes»
688         «"\t\t" + a.name + "= request.json['" + a.name + "']"»
689     «ENDFOR»
690
691     «FOR a : c.attributes»
692         «"\t\t" + c.name.toLowerCase + "_query." + a.name + " = " + a.name»
693     «ENDFOR»
694
695     «"\t\t"»«c.name.toLowerCase»_query.save()
696
697     «"\t\t"»result = «c.name.toLowerCase»_schema.dump(«c.name.toLowerCase»_query)
698     «"\t\t"»return JSONEncoder().encode(result)
699
700     '''
```

**Figura 3-9:** Extracto del código Xtend del generador de código implementado

Fuente: Núñez, Gilberth, 2021

### 3.5 Ejemplificación de la propuesta de DSL

A continuación, se ejemplifica el uso del DSL a través de un escenario extracto de la aplicación modelo el cual contiene tres entidades los cuales son: PERSONA, TITULO y PERSONATITULO.

Este DSL está compuesto por 22 palabras reservadas las cuales son: DBMS, postgresql, mongodb, username, password, host, port, database, class, string, integer, double, boolean, date, float, autoincrement, nullable, unique, primary\_key, ForeignKey, True, False.

```
ejemplo.mydsl
DBMS postgresql{
  username=postgres
  password=postgres
  host=localhost
  port=5432
  database=Prueba
}

class PERSONA{
  IdCedula (string, primary_key=True, nullable=False, unique=True)
  apellidos (string, nullable=False)
  nombre (string, nullable=False)
  usuario (string, unique=True, nullable=False)
  contrasena (string, nullable=False)
  email (string, unique=True, nullable=False)
}

class TITULO{
  IdTitulo (integer, autoincrement=True, primary_key=True, nullable=False)
  titulo(string, nullable=False)
  detalletitulo (string, nullable=False)
}

class PERSONATITULO{
  idPersonaTitulo(integer, autoincrement=True, primary_key=True, nullable=False)
  IdPersona_ref(string, ForeignKey(PERSONA = PERSONA.IdCedula))
  IdTitulo_ref(integer, nullable=False, ForeignKey(TITULO=TITULO.IdTitulo))
}
```

**Figura 3-10:** Ejemplo de gramática del DSL

**Fuente:** Núñez, Gilberth, 2021

Según el extracto de código lo mostrado en la figura 3-10 el DSL debe generar una clase en el archivo “*models*”, en el cual se declara la clase y sus atributos los cuales serán comunicados a la base de datos a través del ORM, también se debe generar su respectiva clase en el archivo de “*serializers*” en el cual se declara la estructura de los datos que serán serializados y enviados, en el archivo “*\_\_init\_\_*” se debe declarar las rutas respectivas por medio de las cuales se podrán acceder los recursos RESTful, se genera un archivo como “PERSONA.py”, “TITULO.py”, “PERSONATITULO.py” los cuales contienen los métodos que trabajaran sus respectivos datos sin un parámetro, finalmente se generó un archivo “idPERSONA.py”, “idTITULO.py”, “idPERSONATITULO.py” los cuales contienen los métodos que trabajaran con sus respectivos datos que necesiten un parámetro.

Las credenciales para la conexión con la base de datos se encuentran en el archivo “Config.py” en este archivo se encuentra los datos que fueron especificados al momento de usar la gramática del DSL.

```

idPERSONA
from flask import request, jsonify
from serializers.Serializers import db, PERSONA, persona_schema
from flask_restful import Resource

class idPERSONA(Resource):
    # GET single PERSONA
    def get(self, idPERSONA):
        persona_query = PERSONA.query.get(idPERSONA)
        return persona_schema.jsonify(persona_query)

    # Update PERSONA
    def put(self, idPERSONA):
        persona_query = PERSONA.query.get(idPERSONA)

        IdCedula= request.json['IdCedula']
        apellidos= request.json['apellidos']
        nombre= request.json['nombre']
        usuario= request.json['usuario']
        contrasena= request.json['contrasena']
        email= request.json['email']

        persona_query.apellidos = apellidos
        persona_query.nombre = nombre
        persona_query.usuario = usuario
        persona_query.contrasena = contrasena
        persona_query.email = email

        db.session.commit()

        return persona_schema.jsonify(persona_query)

    # DELETE PERSONA
    def delete(self, idPERSONA):
        persona_query = PERSONA.query.get(idPERSONA)
        db.session.delete(persona_query)
        db.session.commit()
        return persona_schema.jsonify(persona_query)

```

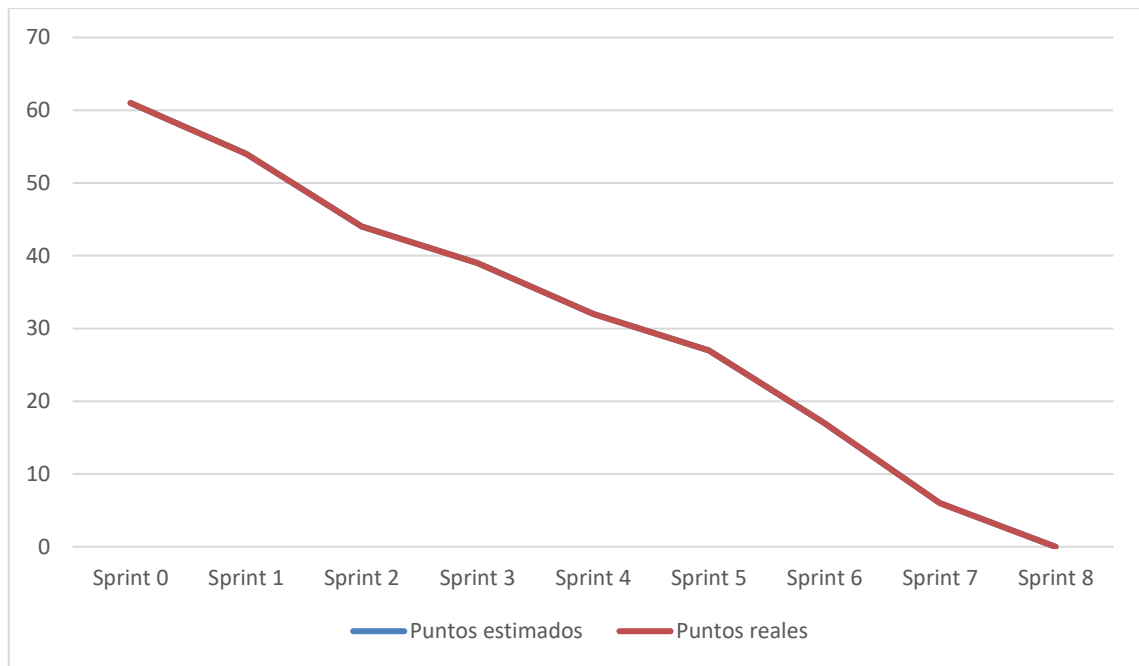
**Figura 3-11:** Extracto del código generado por el DSL

Fuente: Núñez, Gilberth, 2021

Los tipos de datos que soporta el DSL son: string, integer, double, boolean, date, float. Las características de los tipos de datos que soporta son: autoincrement, nullable, unique, primary\_key los cuales deben ir asignados con un valor sea este True o False. Otra característica que soporta el DSL es la característica *ForeignKey* la cual en su declaración debe ir acompañado por el nombre de la clase y el atributo al que se hace referencia.

### 3.5.1 Gestión del proyecto

Para la gestión del proyecto se usó Burn Down Chart, el cual es una representación gráfica del seguimiento del desarrollo de un proyecto. El objetivo de este grafico es exponer a través de las iteraciones la disminución de esfuerzo para cumplir con el desarrollo del proyecto, este grafico nos permitirá identificar si el proyecto terminara o no en el tiempo planificado, en el grafico 1-3 se muestra el seguimiento del proyecto, los *sprints* están detallados en el eje X y en el eje Y de detalla la cantidad de esfuerzo



**Gráfico 1-3:** Gestión del proyecto

**Realizado por:** Núñez, Gilberth, 2021

### 3.5.2 *Gestión de los riesgos*

Las estrategias que se plantearon en la planificación de la gestión de los riesgos anularon los problemas que dichos riesgos podrían causar al presentarse en el desarrollo del DSL, de esta forma el problema de una estimación de tamaño significativamente bajo fue solventado al agregar un margen de error a dicha estimación, los otros posibles riesgos no fueron un problema debido a las buenas estrategias planteadas.

## CAPÍTULO IV

### 4 MARCO DE RESULTADOS

En este capítulo se analizó los resultados obtenidos para evaluar la mejora en la productividad en el desarrollo de software para ello se tomó en cuenta el número de LOCs y el tiempo necesario para generar dichas LOCs, estos valores son comúnmente usados por ingenieros para analizar la productividad del desarrollo (Humphrey, 1997, p. 114).

#### 4.1 Evaluar la mejora de la productividad al utilizar el DSL

##### 4.1.1 Estudio 1: Conexión a una base de datos relacional

Para realizar el primer estudio se planteó un proyecto académico el cual constó de ocho clases las cuales fueron: persona, titulo, persona título, periodo, materia, matricula, nota y pagos, así como también se definió el tipo de base de datos relacional y sus respectivas credenciales para poder realizar la conexión.

Como se expresó en el ejemplo, el DSL consta de 22 palabras reservadas, las cuales fueron usadas para el desarrollo de este primer estudio, estas palabras reservadas nos sirvieron para declarar el tipo de base de datos, así como las credenciales que permitieron la conexión con la base de datos, dichas palabras reservadas fueron *DBMS* para instanciar el gestor de base de datos que se utilizó en este estudio, utilizamos la palabra reservada *postgresql* para identificar el tipo de base de datos, la declaración de las credenciales se debe realizar entre llaves, las palabras reservadas: *username*, *password*, *host*, *port* y *database* se les debe asignar los datos con los cuales se realizara la conexión, de igual forma estos datos se encuentran en el archivo generado "*config.py*" y en caso de ser necesario estos datos podrán ser modificados.

Para la declaración de las clases se utilizó la palabra reservada *class*, en estas clases se declaró los atributos que poseen cada una, la declaración de dichos atributos debe ir entre llaves.

Los atributos a su vez poseen un tipo de dato, así como sus características, el tipo de dato es declarado entre paréntesis al igual que sus características salvo que el tipo de dato debe ser el primer elemento en ser declarado en este estudio se observa el uso de los siguientes tipos de datos: string, integer, date, float.

Las características utilizadas en este estudio fueron: *primary\_key*, *nullable*, *unique* y *foreignkey*, las cuales constan como palabras reservadas de ellas las tres primeras con su respectivo valor de asignación que puede ser *true* o *false*, estos valores forman parte de las 22 palabras reservadas, para el caso de la característica *foreignkey* sus valores de asignación debe ir entre paréntesis, estos valores son la clase y un atributo de la clase como se puede observar en la clase *PERSONATITULO* en el atributo *IdPersona\_ref* usa la clave foránea la cual su estructura es la siguiente *foreignKey(Titulo=TITULO.IdTitulo)*

```
*ejemp.dsl
DBMS postgresql{
  username=postgres
  password=postgres
  host=localhost
  port=5432
  database=Prueba
}

class PERSONA{
  IdCedula (string, primary_key=True, nullable=False, unique=True)
  apellidos (string, nullable=False)
  nombre (string, nullable=False)
  usuario (string, unique=True, nullable=False)
  contrasena (string, nullable=False)
  email (string, unique=True, nullable=False)
}

class TITULO{
  IdTitulo (integer, autoincrement=True, primary_key=True, nullable=False)
  titulo(string, nullable=False)
  detalletitulo (string, nullable=False)
}

class PERSONATITULO{
  IdPersonaTitulo(integer, autoincrement=True, primary_key=True, nullable=False)
  IdPersona_ref(string, ForeignKey(PERSONA = PERSONA.IdCedula))
  IdTitulo_ref(integer, nullable=False, ForeignKey(TITULO=TITULO.IdTitulo))
}

class PERIODO{
  IdPeriodo(integer, autoincrement=True, primary_key=True, nullable=False)
  fechaInicio(date, nullable=False)
  fechaFin (date, nullable=False)
  detalle (string, nullable=False)
}

class MATERIA{
  IdMateria(integer, autoincrement=True, primary_key=True, nullable=False)
  prerrequisito(string)
  correquisito (string)
  detalle (string)
}

class MATRICULA{
  IdMatricula(integer, autoincrement=True, primary_key=True, nullable=False)
  estado(string, nullable=False)
  tipoMatricula (string, nullable=False)
  IdCodigoEstudiante_ref (string, nullable=False, ForeignKey(PERSONA=PERSONA.IdCedula))
  IdMateria_ref(integer, nullable=False, ForeignKey(MATERIA=MATERIA.IdMateria))
  IdPeriodo_ref(integer, nullable=False, ForeignKey(PERIODO=PERIODO.IdPeriodo))
  IdDocente_ref(string, nullable=False, ForeignKey(PERSONA=PERSONA.IdCedula))
}

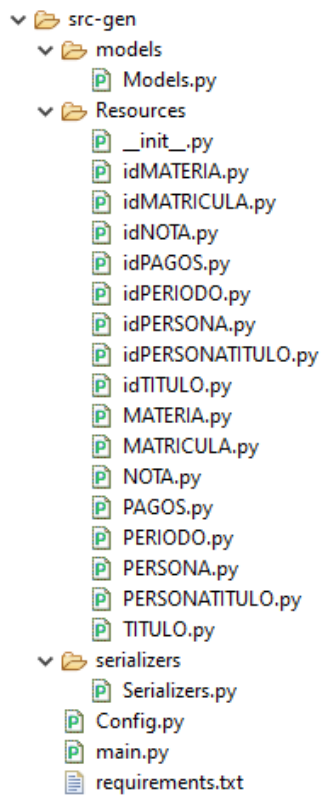
class NOTA{
  IdNotas(integer, autoincrement=True, primary_key=True, nullable=False)
  detalle(string, nullable=False)
  calificacion(float, nullable=False)
  IdMatricula_ref(integer, nullable=False, ForeignKey(MATRICULA=MATRICULA.IdMatricula))
}

class PAGOS{
  IdPagos( integer, autoincrement=True, primary_key=True, nullable=False)
  detalle( string, nullable=False)
  monto( float, nullable=False)
  IdMatricula_ref(integer, nullable=False, ForeignKey(MATRICULA=MATRICULA.IdMatricula))
}
```

**Figura 4-1:** Ejemplo de gramática DSL con conexión a una BD relacional

Fuente: Núñez, Gilberth, 2021

Con el uso de la gramática se generaron 21 archivos de código fuente y un archivo de texto, estos archivos se generan bajo la carpeta *src-gen* como se muestra en la siguiente figura



**Figura 4-2:** Archivos generados a través de la gramática utilizada

**Fuente:** Núñez, Gilberth, 2021

Con el ejemplo de gramática realizado (Figura 4-1) se generó automáticamente 761 líneas de código a partir de 58 LOCs escritas en el DSL.

La tabla 1-4 muestra las líneas de código generadas en los archivos que se encuentran listados, en la cual se observa una tasa de automatización del 13,12 LOC por cada LOC escrita en este DSL (13,12:1)

**Tabla 1-4:** Contabilización de código fuente generado a partir de 59 LOC

Archivo Generado	LOC
Config.py	6
main.py	11
Serializers.py	102

**Tabla 2-4:** Contabilización de código fuente generado a partir de 59 LOC (Continuación de la tabla anterior)

<b>Archivo Generado</b>	<b>LOC</b>
Models.py	256
__init__.py	35
idMATERIA.py	22
idMATRICULA.py	28
idNOTA.py	22
idPAGOS.py	22
idPERIODO.py	22
idPERSONA.py	27
idPERSONATITULO.py	20
idTITULO.py	20
MATERIA.py	20
MATRICULA.py	26
NOTA.py	20
PAGOS.py	20
PERIODO.py	20
PERSONA.py	26
PERSONATITULO.py	18
TITULO.py	18
Total	761

Fuente: Núñez, Gilberth, 2021

Como un índice referente para medir la productividad se tomó en cuenta los valores mostrados en la siguiente tabla.

**Tabla 3-4:** Número de LOCs/hora

<b>Autor</b>	<b>LOCs/hora</b>
Daniel Roy	20 LOC/hr
Baheti et al.	10 LOC/hr
Gwak & Jang	13,3 LOC/hr
Media	14,43 LOC/hr

Realizado por: Núñez, Gilberth, 2021

Fuente: (Roy, 1994, p. 95); (Baheti et al., 2002, p. 6); (Gwak & Jang, 2006)



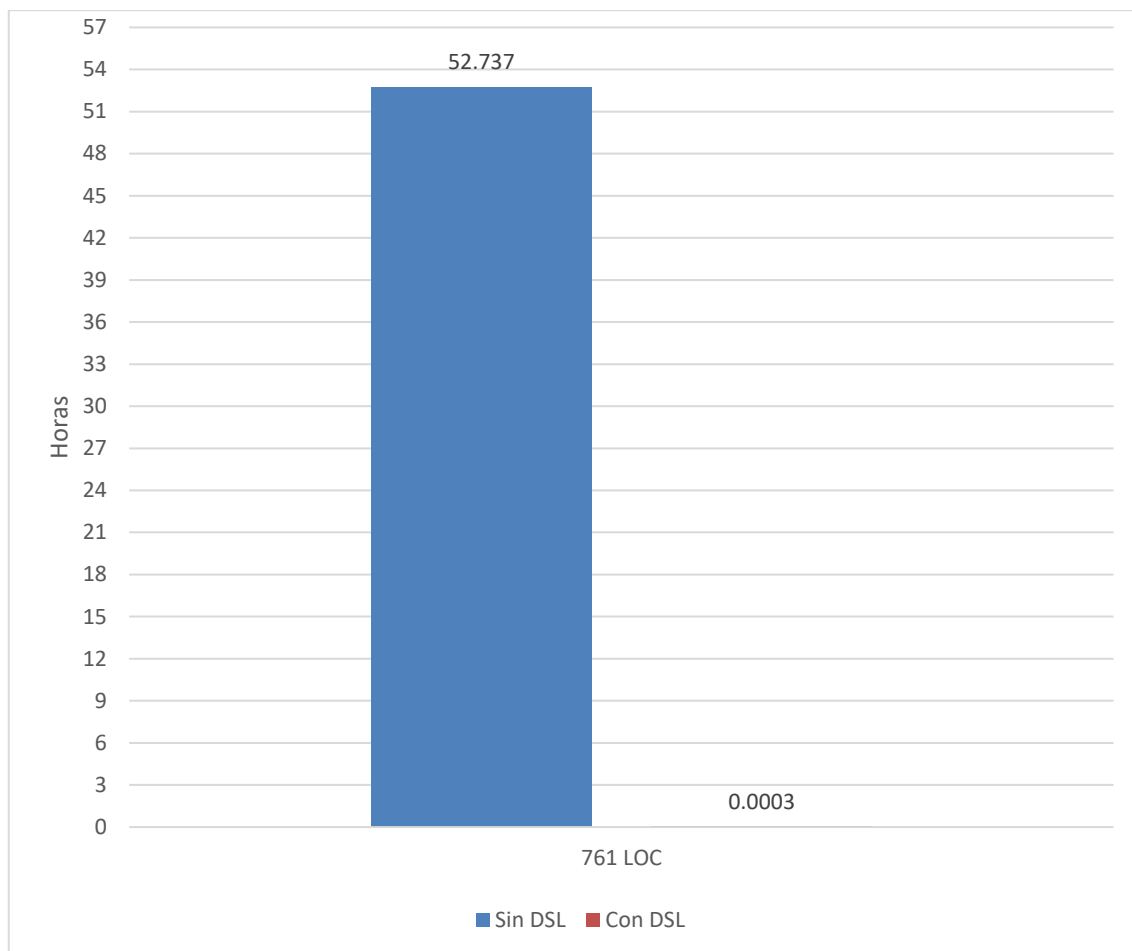
Se tomó en cuenta la media de estos tres valores como una productividad media de 14,43 LOC/hr para la evaluación de este DSL. Con este valor establecemos que las 761 LOCs generadas por el DSL requiere 52,737 horas, un total de 7 días laborales.

#### *4.1.1.1 Análisis de productividad*

Con el uso del DSL el tiempo en generar los archivos con el código mostrados en la tabla 1-4 fue de 0,78 segundos, para este ejemplo se estimó un rendimiento de 3.512.307 LOC/hr. Se observa que el DSL es 243.403 veces más productivo (243.403:1), lo cual representa un aumento de la productividad del 24.340.311%.

#### *4.1.1.2 Análisis de costo*

El salario mínimo de un programador junior de software según el ministerio de trabajo para el 2021 es de \$426,20(MINISTERIO DEL TRABAJO, 2021, p. 51), para el cálculo del costo del proyecto se asume un valor de \$2,66 la hora y un esfuerzo de 52,74 horas, el desarrollo del código fuente mostrado en la tabla 1-4 tendrá un costo de \$140,29. Con el DSL propuesto, el costo de producir el mismo código es aproximadamente un centavo de dólar. El costo de producir el mismo código con el DSL propuesto es 14029 veces más barato.



**Gráfico 1-4:** Histograma del tiempo de desarrollo del proyecto con BD relacional

Realizado por: Núñez, Gilberth, 2021

#### 4.1.2 Estudio 2: Conexión a una base de datos no relacional

Para realizar el segundo estudio se planteó un proyecto académico el cual constó de ocho clases las cuales fueron: persona, título, persona título, periodo, materia, matricula, nota y pagos, así como también se definió el tipo de base de datos no relacional y sus respectivas credenciales para poder realizar la conexión.

Como se expresó en el ejemplo, el DSL consta de 22 palabras reservadas, las cuales fueron usadas para el desarrollo de este segundo estudio, estas palabras reservadas nos sirvieron para declarar el tipo de base de datos, así como las credenciales que permitieron la conexión con la base de datos, dichas palabras reservadas fueron *DBMS* para instanciar el gestor de base de datos que se utilizó en este estudio, utilizamos la palabra reservada *mongodb* para identificar el tipo de base de datos, la declaración de las credenciales se debe realizar entre llaves, las palabras reservadas: *username*, *password*, *host*, *port* y *database* se les debe asignar los datos con los cuales se realizara

la conexión, de igual forma estos datos se encuentran en el archivo generado “*config.py*” y en caso de ser necesario estos datos podrán ser modificados.

Para la declaración de las clases se utilizó la palabra reservada *class*, en estas clases se declaró los atributos que poseen cada una, la declaración de dichos atributos debe ir entre llaves.

Los atributos a su vez poseen un tipo de dato, así como sus características, el tipo de dato es declarado entre paréntesis al igual que sus características con la diferencia que el tipo de dato debe ser el primer elemento en ser declarado en este estudio se observa el uso de los siguientes tipos de datos: string, integer, date, float.

La característica utilizada en este estudio fue: *unique*, la cual consta como palabra reservada con su respectivo valor de asignación que puede ser *true* o *false*, estos valores forman parte de las 22 palabras reservadas.

```
ejemp.dsl
DBMS mongodb{
  username=mongo
  password=sistemas
  host=localhost
  port=5432
  database=Prueba
}

class PERSONA{
  IdCedula (string, unique=True)
  apellidos (string, unique=True)
  nombre (string)
  usuario (string, unique=True)
  contrasena (string)
  email (string, unique=True)
}

class TITULO{
  IdTitulo (integer)
  titulo(string)
  detalletitulo (string)
}

class PERSONATITULO{
  idPersonaTitulo(integer)
  IdPersona_ref(string)
  IdTitulo_ref(integer)
}

class PERIODO{
  IdPeriodo(integer)
  fechaInicio(date)
  fechaFin (date)
  detalle (string)
}

class MATERIA{
  IdMateria(integer)
  prerequisito(string)
  correquisito (string)
  detalle (string)
}

class MATRICULA{
  IdMatricula(integer)
  estado(string)
  tipoMatricula (string)
  IdCodigoEstudiante_ref (string)
  IdMateria_ref(integer)
  IdPeriodo_ref(integer)
  IdDocente_ref(string)
}

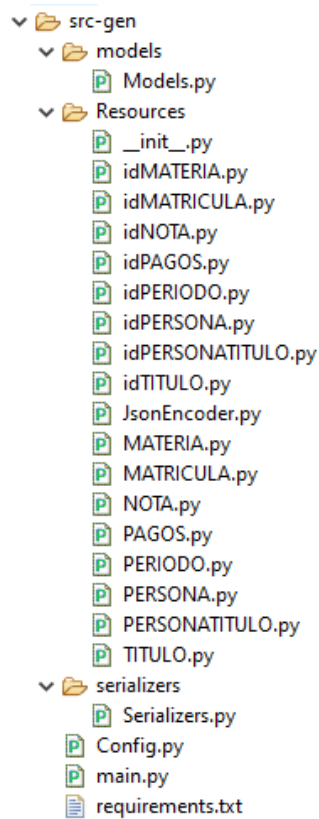
class NOTA{
  IdNotas(integer, unique=True)
  detalle(string)
  calificacion(float)
  IdMatricula_ref(integer)
}

class PAGOS{
  IdPagos( integer)
  detalle( string)
  monto( float)
  IdMatricula_ref(integer)
}
```

**Figura 4 3:** Ejemplo de gramática DSL con conexión a una BD no-relacional

**Fuente:** Núñez, Gilberth, 2021

Con el uso de la gramática se generaron 22 archivos de código fuente y un archivo de texto, estos archivos se generan bajo la carpeta *src-gen* como se muestra en la siguiente figura.



**Figura 4 4:** Archivos generados a través de la gramática utilizando conexión no relacional

Fuente: Núñez, Gilberth, 2021

Este proyecto generó automáticamente 611 líneas de código a partir de 58 LOCs escritas en el DSL.

La tabla 4-4 muestra las líneas de código generadas en los archivos que se encuentran listados, en la cual se observa una tasa de automatización del 10,53 LOC por cada LOC escrita en este DSL (10,53:1)

**Tabla 4-4:** Contabilización de código fuente generado a partir de 58 LOC

Archivo Generado	LOC
Config.py	7
main.py	9
Serializers.py	94
Models.py	51
__init__.py	35
idMATERIA.py	27

**Tabla 5-4:** Contabilización de código fuente generado a partir de 58 LOC (Continuación de la tabla anterior)

<b>Archivo Generado</b>	<b>LOC</b>
idMATRICULA.py	33
idNOTA.py	27
idPAGOS.py	27
idPERIODO.py	27
idPERSONA.py	27
idPERSONATITULO.py	25
idTITULO.py	25
MATERIA.py	23
MATRICULA.py	29
NOTA.py	23
PAGOS.py	23
PERIODO.py	23
PERSONA.py	27
PERSONATITULO.py	21
TITULO.py	21
JsonEncoder.py	7
<b>Total</b>	<b>611</b>

Fuente: Núñez, Gilberth, 2021

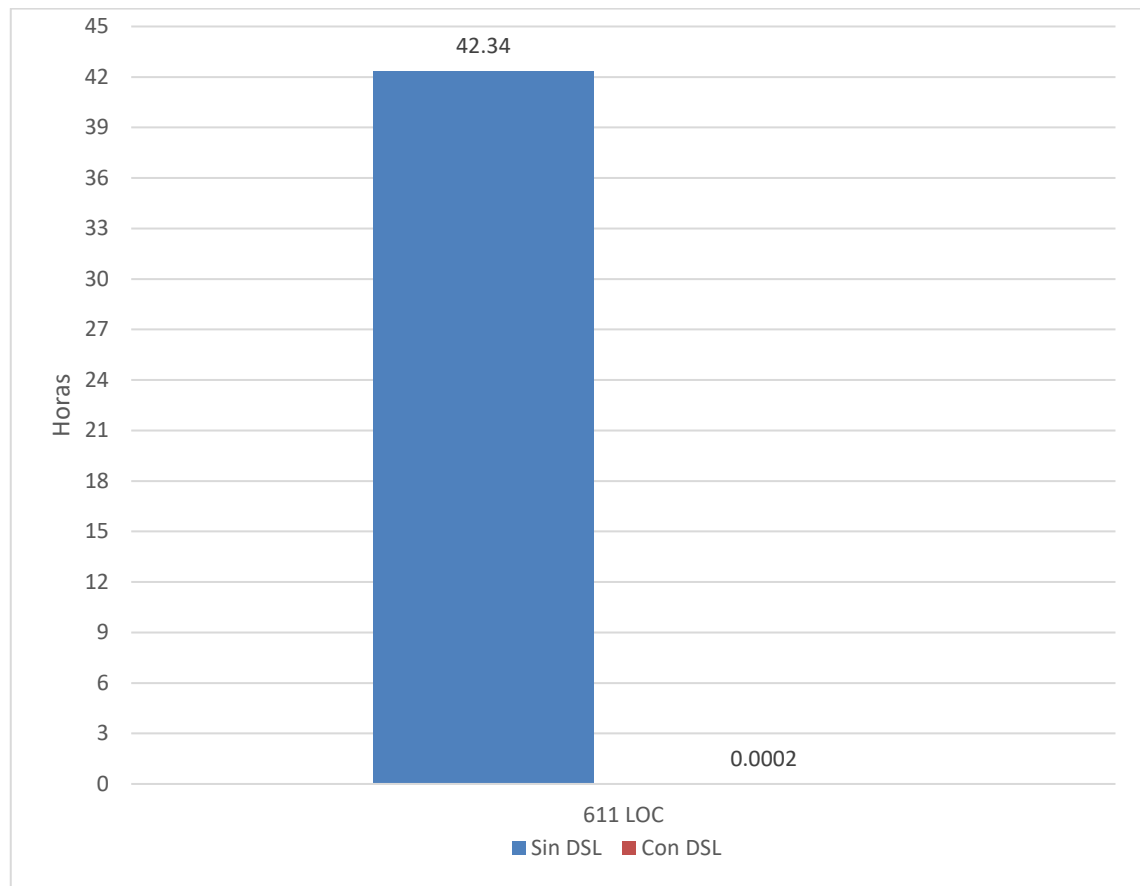
Haciendo uso de la media de 14,43 LOC/hr que se calculó previamente en la tabla 3-4, se midió la mejora de la productividad con el proyecto conectado a una base de datos no relacional. Con el valor 14,43 LOC/hr establecemos que las 611 LOCs generadas por el DSL requiere 42,34 horas, un total de 6 días laborales.

#### 4.1.2.1 *Análisis de productividad*

Con el uso del DSL el tiempo en generar los archivos con el código mostrados en la tabla 1-4 fue de 0,76 segundos, para este ejemplo se estimó un rendimiento de 2.894.210LOC/hr. Se observa que el DSL es 200.568 veces más productivo (200.568:1), lo cual representa un aumento de la productividad del 20.056.897%.

#### 4.1.2.2 Análisis de costo

El salario mínimo de un programador junior de software según el Ministerio de Trabajo para el 2021 es de \$426,20 (MINISTERIO DEL TRABAJO, 2021, p. 51), para el cálculo del costo del proyecto se asume un valor de \$2,66 la hora y un esfuerzo de 42,34 horas, el desarrollo del código fuente mostrado en la tabla 2-4 tendrá un costo de \$112,62. Con el DSL propuesto, el costo de producir el mismo código es aproximadamente un centavo de dólar. El costo de producir el mismo código con el DSL propuesto es 11.262 veces más barato.



**Gráfico 2-4:** Histograma del tiempo de desarrollo del proyecto con BD no-relacional

**Realizado por:** Núñez, Gilberth, 2021

#### 4.1.3 Análisis entre ambos estudios

Se observa un número mayor de LOCs en proyectos con conexión a una base de datos relacional, esto se debe a que los atributos de las clases poseen varias características y esto conlleva a un número mayor de LOCs generadas, así como también una estructura de la base de datos bien definida. En los proyectos con conexión a una base de datos no relacional se observó un número menor de LOCs debido a que estas bases de datos no son tan restrictivas con las características de los atributos de sus clases.

## CONCLUSIONES

- Para el desarrollo del proyecto se utilizaron las herramientas IDE Eclipse y Pycharm, como lenguaje de programación Python, los gestores de base de datos PostgreSQL y MongoDB, el microframework Flask, los cuales posterior a su análisis se concluye que dichas herramientas son de gran importancia técnica ya que gracias a ellas fue posible realizar el DSL y facilitar el proceso de desarrollo.
- Para el desarrollo del DSL se usó la metodología propuesta por Gómez et al., se concluye que dicha metodología facilitó el desarrollo del DSL propuesto, ya que en sus etapas se define claramente los procesos que deben ser llevados a cabo para realizar un proyecto de este tipo.
- Posterior al desarrollo de la aplicación modelo se concluye que dicha aplicación nos permitió identificar los elementos que son utilizados generalmente en un proyecto de desarrollo de software y abstraer dichos elementos para generar la gramática del DSL.
- Definida la gramática del DSL se concluye que el uso de dicha gramática permite la generación de proyectos con conexiones a bases de datos relacionales y no relacionales.
- Para la fase de evaluación de la productividad se tomó en cuenta el número del LOCs y el tiempo medido en horas para generar dichas LOCs, estos valores fueron tomados de otras investigaciones para establecer el valor medio de la productividad con el cual se pudo observar un aumento en la productividad de 20.056.897% en proyectos con conexiones a bases de datos no relacionales y de un 24.340.311% con conexión a bases de datos relacionales.
- Para evaluar la reducción en el costo de desarrollo se tomó el valor establecido por el Ministerio de Trabajo del Ecuador, en el cual se observó una reducción del costo para producir código con el DSL, dicho valor fue 11.262 veces más barato en proyectos con bases de datos no relacionales y 14.029 veces más barato en proyectos con bases de datos relacionales.
- Se concluye finalmente que cada objetivo planteado se cubrió en su totalidad



## RECOMENDACIONES

- Se recomienda el uso de herramientas como el IDE Eclipse, frameworks como Xtext y el compilador Xtend para el desarrollo de nuevos DSL.
- Para el desarrollo del DSL se utilizó la metodología propuesta por Gómez et al. la cual define claramente las etapas para el desarrollo de un proyecto de este tipo, por lo que se recomienda el uso de esta metodología para futuros proyectos
- Para llevar a cabo la parte de gestión del proceso de desarrollo se utilizó la metodología SCRUM la cual permitió cumplir todos los objetivos en el tiempo establecido por lo que se recomienda utilizar esta metodología para futuros proyectos
- Se recomienda continuar con investigaciones para mejorar aspectos de seguridad y modularidad de tal forma que este proyecto mejore continuamente
- Se recomienda que para el uso del DSL se debe tomar en cuenta los tipos de datos que soporta dicha herramienta.

## BIBLIOGRAFÍA

**AGGARWAL, Shalabh.** *Flask Framework Cookbook: Over 80 proven recipes and techniques for Python web development with Flask, 2nd Edition.* Packt Publishing Ltd. 2019. ISBN 978-1-78995-316-9. Google-Books-ID: iF2jDwAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=iF2jDwAAQBAJ&lpg=PP1&dq=iF2jDwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**AHMAD, Haafiz Waheed-ud-din.** *Building RESTful Web Services with PHP 7.* Packt Publishing Ltd., 2017. ISBN 978-1-78728-364-0. Google-Books-ID: xJIGDwAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=xJIGDwAAQBAJ&lpg=PP1&dq=xJIGDwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**BALACHANDAR, Bogunuva Mohanram.** *RESTful Java Web Services: A pragmatic guide to designing and building RESTful APIs using Java.* Packt Publishing Ltd., 2017. ISBN 978-1-78829-496-6. Google-Books-ID: ekFPDwAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=ekFPDwAAQBAJ&lpg=PP1&dq=ekFPDwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**BETTINI, Lorenzo.** *Implementing Domain-Specific Languages with Xtext and Xtend.* Packt Publishing Ltd., 2016. ISBN 978-1-78646-327-2. Google-Books-ID: NkrWDQAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=NkrWDQAAQBAJ&lpg=PP1&dq=NkrWDQAAQBAJ&pg=PP1#v=onepage&q&f=false>

**BURNETTE, Ed.** *Eclipse IDE Pocket Guide: Using the Full-Featured IDE.* O'Reilly Media, Inc., 2005. ISBN 978-0-596-55343-2. Google-Books-ID: MUQyvAIn5YQC. Disponible en: <https://books.google.com.ec/books?id=MUQyvAIn5YQC&lpg=PP1&dq=MUQyvAIn5YQC&pg=PP1#v=onepage&q&f=false>

**CASTRO, Virna Edith Gil, ARMENTA, Xantipa Rosas and QUIROZ, Marisol Olmeda.**

*Fundamentos de las tecnologías de Información: viviendo en una sociedad tecnológica. Volumen I.*

Editorial Digital del Tecnológico de Monterrey., 2014. ISBN 978-607-501-279-7. Google-Books-ID:

vUJuDQAAQBAJ. Disponible en:

<https://books.google.com.ec/books?id=vUJuDQAAQBAJ&lpg=PP1&dq=vUJuDQAAQBAJ&pg=PP1#v=onepage&q&f=false>

**CHODOROW, Kristina.** *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage.*

O'Reilly Media, Inc., 2013. ISBN 978-1-4493-4482-5. Google-Books-ID: uGUKiNkKRJ0C. Disponible

en:

<https://books.google.com.ec/books?id=uGUKiNkKRJ0C&lpg=PP1&dq=uGUKiNkKRJ0C&pg=PP1#v=onepage&q&f=false>

**COBOS, Manuel.** *Un lenguaje de dominio específico para la definición de métricas para lenguajes de modelado.* Escuela Politécnica Superior Universidad Autónoma de Madrid., 2017.

**COPELAND, Rick.** *Essential SQLAlchemy.* O'Reilly Media, Inc., 2008. ISBN 978-1-4493-9084-6.

Google-Books-ID: septpU7dELIC. Disponible en:

<https://books.google.com.ec/books?id=septpU7dELIC&lpg=PP1&dq=septpU7dELIC&pg=PP1#v=onepage&q&f=false>

**COPPERWAITE, Matt and LEIFER, Charles.** *Learning Flask Framework.* Packt Publishing Ltd.,

2015. ISBN 978-1-78398-337-7. Google-Books-ID: HPGoCwAAQBAJ. Disponible en:

<https://books.google.com.ec/books?id=HPGoCwAAQBAJ&lpg=PP1&dq=HPGoCwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**COVANTEC R.L, 2.2.** Librería SQLAlchemy. [online]. 2019. [Accessed 10 August 2020]. Disponible en:

<https://entrenamiento-frameworks-web-python.readthedocs.io/es/latest/leccion2/sqlalchemy.html>

**CUEVAS, Alberto.** *Programar con Python 3.* Lulu.com., 2019. ISBN 978-0-359-68516-5. Google-Books-ID: e4K4DwAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=e4K4DwAAQBAJ&lpg=PP1&dq=e4K4DwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**DI GREGORIO, Federico, VARRAZZO, Daniele and THE PSYCOPG TEAM.** *Psycopg – PostgreSQL database adapter for Python — Psycopg 2.8.7.dev0 documentation.* [online]. 2020. [Accessed 30 December 2020]. Disponible en: <https://www.psycopg.org/docs/>

**DILLON, Tharam.** *Artificial Intelligence in Theory and Practice IV.* Springer., 2015. ISBN 978-3-319-25261-2. Google-Books-ID: lxWcCgAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=lxWcCgAAQBAJ&lpg=PR4&dq=ISBN%20978-3-319-25261-2.&pg=PR4#v=onepage&q=ISBN%20978-3-319-25261-2.&f=false>

**DIXIT, P.K.** *Android.* Vikas Publishing House., 2014. ISBN 978-93-259-7788-4. Google-Books-ID: qKFDDAAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=qKFDDAAAQBAJ&lpg=PP1&dq=qKFDDAAAQBAJ&pg=PP1#v=onepage&q&f=false>

**DOUGLAS, Korry and DOUGLAS, Susan.** *PostgreSQL: A Comprehensive Guide to Building, Programming, and Administering PostgreSQL Databases.* Sams Publishing., 2003. ISBN 978-0-7357-1257-7. Google-Books-ID: gkQVL9pyFVYC. Disponible en: <https://books.google.com.ec/books?id=gkQVL9pyFVYC&lpg=PP1&dq=gkQVL9pyFVYC&pg=PP1#v=onepage&q&f=false>

**DUGGAN, Dominic.** *Enterprise Software Architecture and Design: Entities, Services, and Resources.* John Wiley & Sons., 2012. ISBN 978-0-470-56545-2. Google-Books-ID: Co2jxaOyNOcC. Disponible en: <https://books.google.com.ec/books?id=Co2jxaOyNOcC&lpg=PP1&dq=Co2jxaOyNOcC&pg=PP1#v=onepage&q&f=false>

**DWYER, Gareth.** *Flask By Example*. Packt Publishing Ltd., 2016 ISBN 978-1-78528-348-2. Google-Books-ID: M7PjCwAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=M7PjCwAAQBAJ&lpg=PP1&dq=M7PjCwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**ECLIPSE FOUNDATION.** Xtext - Language Engineering Made Easy! [online]. 8 March 2020a. [Accessed 8 March 2020]. Disponible en: <https://www.eclipse.org/Xtext/index.html#intro-quotes>

**ECLIPSE FOUNDATION.** Xtend - Documentation. [online]. 8 March 2020b. [Accessed 8 March 2020]. Disponible en: <https://www.eclipse.org/xtend/documentation/index.html>

**FERNANDEZ, Arturo.** *Python 3 al descubierto - 2a ed.* Alfaomega Grupo Editor., 2013a. ISBN 978-607-622-008-5. Google-Books-ID: f4BNDAAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=f4BNDAAAQBAJ&lpg=PP1&dq=f4BNDAAAQBAJ&pg=PP1#v=onepage&q&f=false>

**FERNANDEZ, Arturo.** *Python 3 al descubierto - 2a ed.* Alfaomega Grupo Editor., 2013b. ISBN 978-607-622-008-5. Google-Books-ID: f4BNDAAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=f4BNDAAAQBAJ&lpg=PP1&dq=f4BNDAAAQBAJ&pg=PP1#v=onepage&q&f=false>

**FOWLER, Martin.** *Domain-Specific Languages*. Pearson Education., 2010. ISBN 978-0-13-139280-9. Google-Books-ID: ri1muolw\_YwC. Disponible en: [https://books.google.com.ec/books?id=ri1muolw\\_YwC&lpg=PP1&dq=%3A%20ri1muolw\\_YwC&pg=PP1#v=onepage&q&f=false](https://books.google.com.ec/books?id=ri1muolw_YwC&lpg=PP1&dq=%3A%20ri1muolw_YwC&pg=PP1#v=onepage&q&f=false)

**GÓMEZ, Omar S., AGUILETA, Antonio, GÓMEZ, Gerzon and AGUILAR VERA, Raúl.** Estudio del Proceso Software Personal (PSP) en un Entorno Académico. *Revista electrónica de Computación, Informática, Biomédica y Electrónica*. 1 May 2014. Vol. 3, p. I.

**GÓMEZ, Omar S., ROSERO MIRANDA, Raul and VERDÍN, Karen.** CRUDyLeaf: A DSL for Generating Spring Boot REST APIs from Entity CRUD Operations. *Cybernetics and Information Technologies*. 29 August 2020. Vol. 20, p. 3–14. DOI 10.2478/cait-2020-0024.

**GONZALES, Luis.** *Desarrollo de una versión multiplataforma del Sistema Inteligente de Ayuda al Diseño*. Universidad Central “Marta Abreu” de Las Villas., 2015.

**GONZÁLEZ, Raúl.** *Python PARA TODOS.*, 2011.

**HEATON, Jeff.** *JSTL: JSP Standard Tag Library Kick Start*. Sams Publishing., 2003. ISBN 978-0-672-32450-5. Google-Books-ID: nKpvsy8mAZMC. Disponible en: <https://books.google.com.ec/books?id=nKpvsy8mAZMC&lpg=PP1&dq=nKpvsy8mAZMC&pg=PP1#v=onepage&q&f=false>

**HILLAR, Gaston C.** *Hands-On RESTful Python Web Services: Develop RESTful web services or APIs with modern Python 3.7, 2nd Edition*. Packt Publishing Ltd., 2018. ISBN 978-1-78953-278-4. Google-Books-ID: zZ2BDwAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=zZ2BDwAAQBAJ&lpg=PP1&dq=zZ2BDwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**HOWS, David, PLUGGE, Eelco, MEMBREY, Peter and HAWKINS, Tim.** *The Definitive Guide to MongoDB: A Complete Guide to Dealing with Big Data Using MongoDB*. Apress., 2013. ISBN 978-1-4302-5821-6. Google-Books-ID: 5u9ZAgAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=5u9ZAgAAQBAJ&lpg=PP1&dq=5u9ZAgAAQBAJ&pg=PP1#v=onepage&q&f=false>

**HUMPHREY, Watts S.** *Introduction to the Personal Software Process*. Addison-Wesley Professional., 1997. ISBN 978-0-201-54809-9.

**JETBRAINS S.R.O.** Install PyCharm - Help | PyCharm. *PyCharm Help* [online]. 2020a.

[Accessed 29 December 2020]. Disponible en:

<https://www.jetbrains.com/help/pycharm/2020.3/installation-guide.html>

**JETBRAINS S.R.O.** Features - PyCharm. *JetBrains* [online]. 2020b. [Accessed 30 December 2020].

Disponible en: <https://www.jetbrains.com/pycharm/features/>

**JUBA, Salahaldin and VOLKOV, Andrey.** *Learning PostgreSQL 10: A beginner's guide to building high-performance PostgreSQL database solutions*. Packt Publishing Ltd., 2017. ISBN 978-1-78847-066-

7. Google-Books-ID: [ZsxPDwAAQBAJ](https://books.google.com/ec/books?id=ZsxPDwAAQBAJ). Disponible en:

<https://books.google.com/ec/books?id=ZsxPDwAAQBAJ&lpg=PP1&dq=ZsxPDwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**KOURAKLIS, John.** *Introducing Delphi ORM: Object Relational Mapping Using TMS Aurelius*. Apress.,

2019. ISBN 978-1-4842-5013-6. Google-Books-ID: [LuWmDwAAQBAJ](https://books.google.com/ec/books?id=LuWmDwAAQBAJ). Disponible en:

<https://books.google.com/ec/books?id=LuWmDwAAQBAJ&lpg=PR1&dq=LuWmDwAAQBAJ&pg=PR1#v=onepage&q&f=false>

**KULKARNI, Ram.** *Java EE 8 Development with Eclipse: Develop, test, and troubleshoot Java Enterprise applications rapidly with Eclipse, 3rd Edition*. Packt Publishing Ltd., 2018. ISBN 978-1-78883-388-2.

Google-Books-ID: [SMNiDwAAQBAJ](https://books.google.com/ec/books?id=SMNiDwAAQBAJ). Disponible en:

<https://books.google.com/ec/books?id=SMNiDwAAQBAJ&lpg=PP1&dq=SMNiDwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**KUMAR, Santosh.** *Spring And Hibernate*. Tata McGraw-Hill Education., 2009. ISBN 978-0-07-007765-

2. Google-Books-ID: [NfNbbhBRcOkC](https://books.google.com/ec/books?id=NfNbbhBRcOkC). Disponible en:

<https://books.google.com/ec/books?id=NfNbbhBRcOkC&lpg=PP1&dq=NfNbbhBRcOkC&pg=PP1#v=onepage&q&f=false>

**LÓPEZ, Ángel.** *Lenguajes específicos de dominio gráficos y textuales: Un estudio comparativo.*, 2013

**LORIA, Steven.** marshmallow: simplified object serialization — marshmallow 3.10.0 documentation. [online]. 2020a. [Accessed 31 December 2020]. Disponible en: <https://marshmallow.readthedocs.io/en/stable/index.html>

**LORIA, Steven.** Why marshmallow? — marshmallow 3.10.0 documentation. [online]. 2020b. [Accessed 31 December 2020]. Disponible en: <https://marshmallow.readthedocs.io/en/stable/why.html>

**MARTELLI, Alex.** *Python in a Nutshell.* O'Reilly Media, Inc., 2003. ISBN 978-0-596-00188-9. Google-Books-ID: 6TEcaEzA8N0C. Disponible en: <https://books.google.com.ec/books?id=6TEcaEzA8N0C&lpg=PP1&dq=6TEcaEzA8N0C&pg=PP1#v=onepage&q&f=false>

**MEHTA, Vijay P.** *Pro LINQ Object Relational Mapping in C# 2008.* Apress., 2008. ISBN 978-1-4302-0597-5. Google-Books-ID: ZbUYAAAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=ZbUYAAAAQBAJ&lpg=PP1&dq=ZbUYAAAAQBAJ&pg=PP1#v=onepage&q&f=false>

**MINISTERIO DEL TRABAJO.** *SALARIOS MÍNIMOS SECTORIALES 2021 ANEXO 1: ESTRUCTURAS OCUPACIONALES - SALARIOS MÍNIMOS SECTORIALES Y TARIFAS* [online]. 2021. Disponible en: <https://www.legalecuador.com/wp-content/uploads/2021/01/legalecuador-anexo-1-comisiones-sectoriales-2021.pdf>

**MONGODB.** Tools — PyMongo 2.7 documentation. [online]. 2014. [Accessed 1 January 2021]. Disponible en: <https://api.mongodb.com/python/2.7/tools.html>

**MONGODB.** La base de datos líder del mercado para aplicaciones modernas. *MongoDB* [online]. 13 March 2020. [Accessed 13 March 2020]. Disponible en: <https://www.mongodb.com/es>. Library Catalog: [www.mongodb.com](http://www.mongodb.com)



**MONTENEGRO, Carlos Enrique, CUEVA, Juan Manuel, MARTINEZ, anjuán Óscar Sanjuán Óscar Sanjuán and GAONA, Paulo Alonso.** Desarrollo de un lenguaje de dominio específico para sistemas de gestión de aprendizaje y su herramienta de implementación “KiwiDSM” mediante ingeniería dirigida por modelos. 2010. Vol. 15, no. 2, p. 67–81.

**MORA, Beatriz, RUIZ, Francisco, GARCÍA, Félix and PIATTINI, Mario.** *Definición de Lenguajes de Modelos MDA vs DSL.* España: UNIVERSIDAD DE CASTILLA-LA MANCHA., 2006.

**NGUYEN, Quan.** *Hands-On Application Development with PyCharm: Accelerate your Python applications using practical coding techniques in PyCharm.* Packt Publishing Ltd., 2019. ISBN 978-1-78934-286-4. Google-Books-ID: zZuzDwAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=zZuzDwAAQBAJ&lpg=PP1&dq=zZuzDwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**OBE, Regina O. and HSU, Leo S.** *PostgreSQL: Up and Running: A Practical Introduction to the Advanced Open Source Database.* O’Reilly Media, Inc., 2014. ISBN 978-1-4493-7317-7. Google-Books-ID: ASjJBQAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=ASjJBQAAQBAJ&lpg=PP1&dq=ASjJBQAAQBAJ&pg=PP1#v=onepage&q&f=false>

**POHJONEN, Risto and KELLY, Steven.** Domain-specific modeling. *Dr. Dobb’s Journal.* 1 August 2002. Vol. 27.

**POSTGRESQL.** PostgreSQL: The world’s most advanced open source database. [online]. 13 March 2020. [Accessed 13 March 2020]. Disponible en: <https://www.postgresql.org/>

**PYPA.** Virtualenv documentation. [online]. 2007. [Accessed 6 January 2021]. Disponible en: <https://virtualenv.pypa.io/en/latest/#>

**RICO, David F.** *ROI of Software Process Improvement: Metrics for Project Managers and Software*

*Engineers*. J. Ross Publishing., 2004. ISBN 978-1-932159-24-0. Google-Books-ID: UQT0AMDZ3w4C. Disponible en: <https://books.google.com.ec/books?id=UQT0AMDZ3w4C&lpg=PP1&dq=UQT0AMDZ3w4C&pg=PP1#v=onepage&q&f=false>

**ROMANO, Fabrizio, HILLAR, Gaston C. and RAVINDRAN, Arun.** *Learn Web Development with Python: Get hands-on with Python Programming and Django web development*. Packt Publishing Ltd., 2018. ISBN 978-1-78995-088-5. Google-Books-ID: KSyBDwAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=KSyBDwAAQBAJ&lpg=PP1&dq=KSyBDwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**SÁNCHEZ, Alberto.** *Aplicación web para el análisis estadístico de datos de la calidad de agua y del aire*. Universidad Internacional de La Rioja(UNIR), 2017.

**SECRETARÍA NACIONAL DE PLANIFICACIÓN Y DESARROLLO - SENPLADES.** *Plan Nacional de Desarrollo 2017 – 2021 Toda una Vida*. 2017.

**SIERRA, Fco Javier Ceballos.** *JAVA. Interfaces gráficas y aplicaciones para Internet. 4ª Edición*. Grupo Editorial RA-MA., 2015. ISBN 978-84-9964-479-0. Disponible en: <https://books.google.com.ec/books?id=A7E-DwAAQBAJ&lpg=PT4&dq=ISBN%20978-84-9964-479-0&pg=PP1#v=onepage&q=ISBN%20978-84-9964-479-0&f=false>

**SMITH, BEN.** *Beginning JSON*. Apress., 2015. ISBN 978-1-4842-0202-9. Google-Books-ID: ZYYnCGAAQBAJ. Disponible en: <https://books.google.com.ec/books?id=ZYYnCGAAQBAJ&lpg=PP1&dq=ZYYnCGAAQBAJ&pg=PP1#v=onepage&q&f=false>

**SQLALCHEMY.** *SQLAlchemy - The Database Toolkit for Python*. [online]. 13 March 2020. [Accessed 13 March 2020]. Disponible en: <https://www.sqlalchemy.org/>

**SRIPARASA, Sai S. and D'MELLO, Bruno Joseph.** *JavaScript and JSON Essentials: Build light weight, scalable, and faster web applications with the power of JSON, 2nd Edition*. Packt Publishing Ltd.,

2018. ISBN 978-1-78862-876-1. Google-Books-ID: Fd9YDwAAQBAJ. Disponible en:  
<https://books.google.com.ec/books?id=Fd9YDwAAQBAJ&lpg=PP1&dq=Fd9YDwAAQBAJ&pg=PP1#v=onepage&q&f=false>

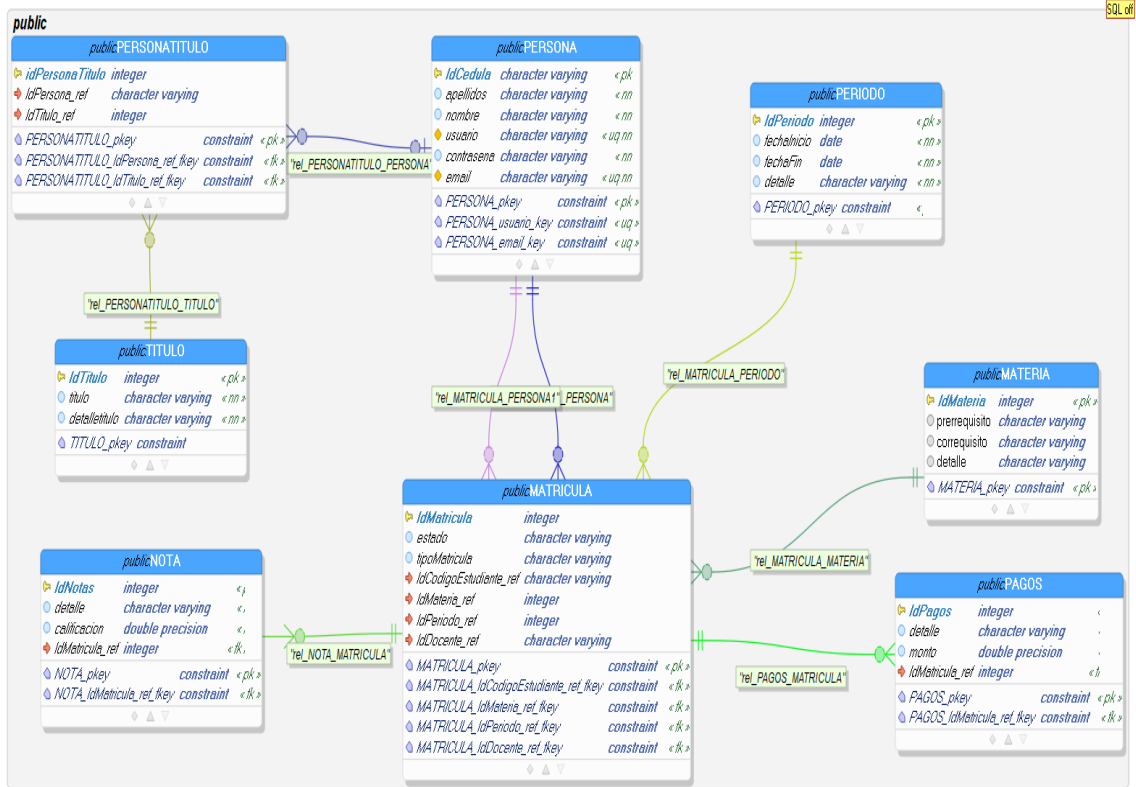
**TUDOR, James.** *Python para principiantes: Aprenda Python en 5 días con orientación paso a paso y ejercicios prácticos.* Babelcube Inc., 2019. ISBN 978-1-07-150926-5. Google-Books-ID: 3huyDwAAQBAJ. Disponible en:  
<https://books.google.com.ec/books?id=3huyDwAAQBAJ&lpg=PP1&dq=3huyDwAAQBAJ&pg=PP1#v=onepage&q&f=false>

**VALENCIA, Carlos.** *Diseño de un prototipo de DSL basado en ingeniería de modelos para el control a distancia de un dron terrestre.* Universidad de las Fuerzas Armadas., 2017.

# ANEXOS

## Anexo A: Base de datos

### Modelo lógico de la base de datos



Fuente: Núñez, Gilberth, 2021

### Diccionario de la base de datos

#### Tabla Materia

public.MATERIA							Table
Name	Data type	PK	FK	UQ	Not null	Default value	Description
IdMateria	integer	✓			✓	nextval('public."MATERIA_IdMateria_seq"'::regclass)	
prerequisito	character varying						
correquisito	character varying						
detalle	character varying						
Constraints							
Name	Type	Column(s)	References	Expression	Description		
MATERIA_pkey	PRIMARY KEY	IdMateria					

Fuente: Núñez, Gilberth, 2021

## Tabla Matricula

pública.MATRICULA							Table
Name	Data type	PK	FK	UQ	Not null	Default value	Description
IdMatricula	integer	✓			✓	nextval('public."MATRICULA_IdMatricula_seq"::regclass)	
estado	character varying				✓		
tipoMatricula	character varying				✓		
IdCodigoEstudiante_ref	character varying		✓		✓		
IdMateria_ref	integer		✓		✓		
IdPeriodo_ref	integer		✓		✓		
IdDocente_ref	character varying		✓		✓		
Constraints							
Name	Type	Column(s)	References	Expression	Description		
MATRICULA_pkey	PRIMARY KEY	IdMatricula					
MATRICULA_IdCodigoEstudiante_ref_fkey	FOREIGN KEY	IdCodigoEstudiante_ref	90pública.PERSON A				
MATRICULA_IdMateria_ref_fkey	FOREIGN KEY	IdMateria_ref	90pública.MATERI A				
MATRICULA_IdPeriodo_ref_fkey	FOREIGN KEY	IdPeriodo_ref	90pública.PERIOD O				
MATRICULA_IdDocente_ref_fkey	FOREIGN KEY	IdDocente_ref	90pública.PERSON A				

Fuente: Núñez, Gilberth, 2021

## Tabla Nota

pública.NOTA							Table
Name	Data type	PK	FK	UQ	Not null	Default value	Description
IdNotas	integer	✓			✓	nextval('public."NOTA_IdNotas_seq"::regclass)	
detalle	character varying				✓		
calificacion	double precision				✓		
IdMatricula_ref	integer		✓		✓		
Constraints							
Name	Type	Column(s)	References	Expression	Description		
NOTA_pkey	PRIMARY KEY	IdNotas					
NOTA_IdMatricula_ref_fkey	FOREIGN KEY	IdMatricula_ref	90pública.MATRICULA				

Fuente: Núñez, Gilberth, 2021

## Tabla Pagos

pública.PAGOS							Table
Name	Data type	PK	FK	UQ	Not null	Default value	Description
IdPagos	integer	✓			✓	nextval('public."PAGOS_IdPagos_seq"::regclass)	
Detalle	character varying				✓		
Monto	double precision				✓		
IdMatricula_ref	integer		✓		✓		
Constraints							
Name	Type	Column(s)	References	Expression	Description		
PAGOS_pkey	PRIMARY KEY	IdPagos					
PAGOS_IdMatricula_ref_fkey	FOREIGN KEY	IdMatricula_ref	90pública.MATRICULA				

Fuente: Núñez, Gilberth, 2021

## Tabla Periodo

pública.PERIODO								Table
Name	Data type	PK	FK	UQ	Not null	Default value	Description	
IdPeriodo	integer	✓			✓	nextval('public."PERIODO_IdPeriodo_seq"::regclass')		
fechalnicio	date				✓			
fechaFin	date				✓			
detalle	character varying				✓			
Constraints								
Name	Type	Column(s)	References	Expression	Description			
PERIODO_pkey	PRIMARY KEY	IdPeriodo						

Fuente: Núñez, Gilberth, 2021

## Tabla Persona

pública.PERSONA									Table
Name	Data type	PK	FK	UQ	Not null	Default value	Description		
IdCedula	character varying	✓			✓				
apellidos	character varying				✓				
nombre	character varying				✓				
usuario	character varying			✓	✓				
contrasena	character varying				✓				
email	character varying			✓	✓				
Constraints									
Name	Type	Column(s)	References	Expression	Description				
PERSONA_pkey	PRIMARY KEY	IdCedula							
PERSONA_usuario_key	UNIQUE	usuario							
PERSONA_email_key	UNIQUE	email							

Fuente: Núñez, Gilberth, 2021

## Tabla PersonaTitulo

pública.PERSONATITULO								Table
Name	Data type	PK	FK	UQ	Not null	Default value	Description	
idPersonaTitulo	integer	✓			✓	nextval('public."PERSONATITULO_idPersonaTitulo_seq"::regclass')		
IdPersona_ref	character varying		✓					
IdTitulo_ref	integer		✓		✓			
Constraints								
Name	Type	Column(s)	References	Expression	Description			
PERSONATITULO_pkey	PRIMARY KEY	idPersonaTitulo						
PERSONATITULO_IdPersona_ref_fkey	FOREIGN KEY	IdPersona_ref	91 pública.PERSONA					
PERSONATITULO_IdTitulo_ref_fkey	FOREIGN KEY	IdTitulo_ref	91 pública.TITULO					

Fuente: Núñez, Gilberth, 2021

## Tabla Titulo

pública.TITULO								Table
Name	Data type	PK	FK	UQ	Not null	Default value	Description	
IdTitulo	integer	✓			✓	nextval('public."TITULO_IdTitulo_seq"::regclass')		
titulo	character varying				✓			
detalletitulo	character varying				✓			
Constraints								
Name	Type	Column(s)	References	Expression	Description			
TITULO_pkey	PRIMARY KEY	IdTitulo						

Fuente: Núñez, Gilberth, 2021

## Anexo B: Tablas de Historias Técnicas

### *HT\_1 Recolección de información*

Historia Técnica 1		
<b>ID:</b> HT_1	<b>Nombre de la Historia:</b> Recolección de información	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 1	
<b>Prioridad:</b> Alta (Alta/Media/Baja)	<b>Puntos Estimados:</b> 2	
	<b>Puntos Reales:</b> 2	
<b>Fecha Inicio:</b> 15/06/2020	<b>Fecha Fin:</b> 19/06/2020	
<b>Descripción:</b> Como desarrollador se desea recolectar información acerca del desarrollo de los DSL		
Tareas de Ingeniería		
ID_TI	Descripción TI	Esfuerzo
TI_1	Recolectar y analizar la información referente al desarrollo del DSL	2

Fuente: Núñez, Gilberth, 2021

### *HT\_2 Selección de las herramientas para el desarrollo*

Historia Técnica 2		
<b>ID:</b> HT_2	<b>Nombre de la Historia:</b> Selección de las herramientas para el desarrollo	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 1	
<b>Prioridad:</b> Alta (Alta/Media/Baja)	<b>Puntos Estimados:</b> 3	
	<b>Puntos Reales:</b> 3	
<b>Fecha Inicio:</b> 22/06/2020	<b>Fecha Fin:</b> 03/07/2020	
<b>Descripción:</b> Como desarrollador se desea seleccionar e instalar las herramientas para el desarrollo del DSL		
Tareas de Ingeniería		
ID_TI	Descripción TI	Esfuerzo
• TI_2	• Analizar las herramientas existentes para el desarrollo del DSL y seleccionar cuales serán utilizadas en el desarrollo de este • Instalar las herramientas software seleccionadas para el desarrollo del DSL	• 1
• TI_3		• 2

Fuente: Núñez, Gilberth, 2021

### *HT\_3 Identificar la estructura general de proyectos desarrollados en Python*

Historia Técnica 3		
<b>ID:</b> HT_3	<b>Nombre de la Historia:</b> Identificar la estructura general de proyectos desarrollados en Python	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 1	
<b>Prioridad:</b> Alta (Alta/Media/Baja)	<b>Puntos Estimados:</b> 2	
	<b>Puntos Reales:</b> 2	
<b>Fecha Inicio:</b> 06/07/2020	<b>Fecha Fin:</b> 10/07/2020	
<b>Descripción:</b> Como desarrollador se desea identificar la estructura general de proyectos desarrollados en el lenguaje Python		
Tareas de Ingeniería		
ID_TI	Descripción TI	Esfuerzo
• TI_4	• Analizar proyectos desarrollados en Python con el uso del micro-framework flask y el uso de ORM para comunicarse con la base de datos • Definir la estructura en la que se generará los proyectos	1
• TI_5		1

Fuente: Núñez, Gilberth, 2021

#### HT\_4 Diseñar la base de datos

Historia Técnica 4		
<b>ID:</b> HT_4	<b>Nombre de la Historia:</b> Diseñar la base de datos	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 2	
<b>Prioridad:</b> Alta (Alta/Media/Baja)	<b>Puntos Estimados:</b> 1	
	<b>Puntos Reales:</b> 1	
<b>Fecha Inicio:</b> 13/07/2020	<b>Fecha Fin:</b> 15/07/2020	
<b>Descripción:</b> Como desarrollador se desea diseñar la base de datos		
Tareas de Ingeniería		
ID_TI	Descripción TI	Esfuerzo
TI_6	Como desarrollador, se realizará un estudio y análisis de la estructura de la base de datos	1

Fuente: Núñez, Gilberth, 2021

#### Anexo B1.5: HT\_5 Definir el estándar de programación

Historia Técnica 5		
<b>ID:</b> HT_5	<b>Nombre de la Historia:</b> Definir el estándar de programación	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 2	
<b>Prioridad:</b> Alta (Alta/Media/Baja)	<b>Puntos Estimados:</b> 1	
	<b>Puntos Reales:</b> 1	
<b>Fecha Inicio:</b> 16/07/2020	<b>Fecha Fin:</b> 17/07/2020	
<b>Descripción:</b> Como desarrollador se desea definir el estándar de programación		
Tareas de Ingeniería		
ID_TI	Descripción TI	Esfuerzo
TI_7	Como desarrollador se desea establecer el estándar de programación bajo el cual se registrará el desarrollo del proyecto	1

Fuente: Núñez, Gilberth, 2021

#### HT\_6 Estudiar el dominio de la aplicación

Historia Técnica 6		
<b>ID:</b> HT_6	<b>Nombre de la Historia:</b> Estudiar el dominio de la aplicación	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 2, 3, 4	
<b>Prioridad:</b> Alta (Alta/Media/Baja)	<b>Puntos Estimados:</b> 8	
	<b>Puntos Reales:</b> 8	
<b>Fecha Inicio:</b> 20/07/2020	<b>Fecha Fin:</b> 11/09/2020	
<b>Descripción:</b> Como desarrollador se desea estudiar el dominio de la aplicación		
Tareas de Ingeniería		
ID_TI	Descripción TI	Esfuerzo
• TI_8	• Como desarrollador se desea estudiar el archivo main del proyecto e identificar sus componentes	•2
• TI_9	• Como desarrollador se desea estudiar el archivo de configuración y conexión a la base de datos	•2
• TI_10	• Como desarrollador se desea estudiar el archivo modelos	•2
• TI_11	• Como desarrollador se desea estudiar el archivo serializadores	•2
• TI_12	• Como desarrollador se desea estudiar el archivo de declaración de los servicios REST	•2
• TI_13	• Como desarrollador se desea estudiar el archivo de servicios REST	•5

Fuente: Núñez, Gilberth, 2021



### HT\_7 Desarrollo de la aplicación modelo

Historia Técnica 7		
<b>ID:</b> HT_7	<b>Nombre de la Historia:</b> Desarrollo de la aplicación modelo	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 4, 5, 6, 7, 8	
<b>Prioridad:</b> (Alta/Media/Baja)	<b>Puntos Estimados:</b> 13	
	<b>Puntos Reales:</b> 13	
<b>Fecha Inicio:</b> 14/09/2020	<b>Fecha Fin:</b> 06/11/2020	
<b>Descripción:</b> Como desarrollador se desea desarrollar la aplicación modelo		
Tareas de Ingeniería		
ID_TI	Descripción TI	Esfuerzo
•TI_14	•Desarrollar el archivo main del proyecto	•2
•TI_15	•Desarrollar el archivo de configuración y conexión a la base de datos	•2
•TI_16	•Desarrollar el archivo modelos	•2
•TI_17	•Desarrollar el archivo serializadores	•2
•TI_18	•Desarrollar el archivo de declaración de los servicios REST	•2
•TI_19	•Desarrollar el archivo de servicios REST	•5

Fuente: Núñez, Gilberth, 2021

### HT\_8 Desarrollar la gramática del lenguaje

Historia Técnica 8		
<b>ID:</b> HT_8	<b>Nombre de la Historia:</b> Desarrollar la gramática del lenguaje	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 9	
<b>Prioridad:</b> Alta (Alta/Media/Baja)	<b>Puntos Estimados:</b> 5	
	<b>Puntos Reales:</b> 5	
<b>Fecha Inicio:</b> 09/11/2020	<b>Fecha Fin:</b> 04/12/2020	
<b>Descripción:</b> Como desarrollador se desea desarrollar la gramática del lenguaje		
Tareas de Ingeniería		
ID_TI	Descripción TI	Esfuerzo
•TI_20	•Como desarrollador se desea definir las reglas gramaticales	•3
•TI_21	•Como desarrollador se desea establecer las reglas gramaticales para bases de datos tipo PostgreSQL	•2
•TI_22	•Como desarrollador se desea establecer las reglas gramaticales para bases de datos tipo MongoDB	•2

Fuente: Núñez, Gilberth, 2021

### HT\_9 Generar el metamodelo

Historia Técnica 9		
<b>ID:</b> HT_9	<b>Nombre de la Historia:</b> Generar el metamodelo	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 10	
<b>Prioridad:</b> Alta (Alta/Media/Baja)	<b>Puntos Estimados:</b> 1	
	<b>Puntos Reales:</b> 1	
<b>Fecha Inicio:</b> 07/12/2020	<b>Fecha Fin:</b> 08/12/2020	
<b>Descripción:</b> Como desarrollador se desea generar el metamodelo		
Tareas de Ingeniería		
ID_TI	Descripción TI	Esfuerzo
•TI_23	•Como desarrollador se desea analizar el proceso de generación del metamodelo	•1
•TI_24	•Como desarrollador se desea generar el metamodelo con la herramienta de Eclipse	•1

Fuente: Núñez, Gilberth, 2021

### **HT\_10 Desarrollar el generador de código**

<b>Historia Técnica 10</b>		
<b>ID:</b> HT_10	<b>Nombre de la Historia:</b> Desarrollar el generador de código	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 10, 11, 12	
<b>Prioridad:</b> Alta (Alta/Media/Baja)	<b>Puntos Estimados:</b> 8	
	<b>Puntos Reales:</b> 8	
<b>Fecha Inicio:</b> 09/12/2020	<b>Fecha Fin:</b> 19/01/2021	
<b>Descripción:</b> Como desarrollador se desea desarrollar el generador de código		
<b>Tareas de Ingeniería</b>		
<b>ID_TI</b>	<b>Descripción TI</b>	<b>Esfuerzo</b>
• TI_25	• Como desarrollador se desea codificar el generador del archivo modelo	• 2
• TI_26	• Como desarrollador se desea codificar el generador del archivo serializadores	• 2
• TI_27	• Como desarrollador se desea codificar el generador del archivo de declaración de los servicios REST	• 2
• TI_28	• Como desarrollador se desea codificar el generador del archivo de servicios REST	• 3
• TI_29	• Como desarrollador se desea codificar el generador del archivo de configuración y main	• 2

Fuente: Núñez, Gilberth, 2021

### **HT\_11 Generar un proyecto con el DSL**

<b>Historia Técnica 11</b>		
<b>ID:</b> HT_11	<b>Nombre de la Historia:</b> Generar un proyecto con el DSL	
<b>Usuario:</b> Desarrollador	<b>Sprint:</b> 12	
<b>Prioridad:</b> Alta (Alta/Media/Baja)	<b>Puntos Estimados:</b> 1	
	<b>Puntos Reales:</b> 1	
<b>Fecha Inicio:</b> 20/01/2021	<b>Fecha Fin:</b> 22/01/2021	
<b>Descripción:</b> Como desarrollador se desea generar un proyecto con el DSL		
<b>Tareas de Ingeniería</b>		
<b>ID_TI</b>	<b>Descripción TI</b>	<b>Esfuerzo</b>
TI_30	Como desarrollador se desea definir la base de datos y las clases para generar un proyecto	1

Fuente: Núñez, Gilberth, 2021

## Anexo C: Tablas de Tareas de Ingeniería

### TI\_1 Recolectar y analizar la información

Tarea	
Numero tarea: TI_1	Número historia: HT_1
Nombre tarea: Recolectar y analizar la información	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 2
Fecha inicio: 15/06/2020	Fecha fin: 18/06/2020
Programador responsable: Gilberth Núñez	
Descripción: Recolectar y analizar la información referente al desarrollo del DSL	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Generar la documentación de la información encontrada</li></ul>	

Fuente: Núñez, Gilberth, 2021

### TI\_2 Analizar las herramientas

Tarea	
Numero tarea: TI_2	Número historia: HT_2
Nombre tarea: Analizar las herramientas	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 1
Fecha inicio: 22/06/2020	Fecha fin: 25/06/2020
Programador responsable: Gilberth Núñez	
Descripción: Analizar las herramientas existentes para el desarrollo del DSL y seleccionar cuales serán utilizadas en el desarrollo de este	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Generar la documentación sobre las herramientas encontradas</li></ul>	

Fuente: Núñez, Gilberth, 2021

### TI\_3 Instalar el software seleccionado

Tarea	
Numero tarea: TI_3	Número historia: HT_2
Nombre tarea: Instalar el software seleccionado	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 2
Fecha inicio: 29/06/2020	Fecha fin: 02/07/2020
Programador responsable: Gilberth Núñez	
Descripción: Instalar las herramientas software seleccionadas para el desarrollo del DSL	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Realizar la instalación de las herramientas</li><li>• Verificar el funcionamiento de las herramientas instaladas</li></ul>	

Fuente: Núñez, Gilberth, 2021

#### TI\_4 Analizar proyectos desarrollados en Python

Tarea	
Numero tarea: TI_4	Número historia: HT_3
Nombre tarea: Analizar proyectos desarrollados en Python	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 1
Fecha inicio: 06/07/2020	Fecha fin: 07/07/2020
Programador responsable: Gilberth Núñez	
Descripción: Analizar proyectos desarrollados en Python con el uso del micro-framework flask y el uso de ORM para comunicarse con la base de datos	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Documentar la estructura encontrada en los diferentes proyectos</li><li>• Verificar que los proyectos encontrados trabajen con las herramientas seleccionadas</li></ul>	

Fuente: Núñez, Gilberth, 2021

#### TI\_5 Definir la estructura en la que se generará los proyectos

Tarea	
Numero tarea: TI_5	Número historia: HT_3
Nombre tarea: Definir la estructura en la que se generará los proyectos	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 1
Fecha inicio: 09/07/2020	Fecha fin: 10/07/2020
Programador responsable: Gilberth Núñez	
Descripción: Definir la estructura en la que se generará los proyectos con el uso del DSL de tal forma que sea claro y fácil de entender	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Documentar la estructura definida para la generación del proyecto</li><li>• Verificar la estructura definida</li></ul>	

Fuente: Núñez, Gilberth, 2021

#### TI\_6 Estudio y análisis de la base de datos

Tarea	
Numero tarea: TI_6	Número historia: HT_4
Nombre tarea: Estudio y análisis de la base de datos	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 1
Fecha inicio: 13/07/2020	Fecha fin: 14/07/2020
Programador responsable: Gilberth Núñez	
Descripción: Como desarrollador, se realizará un estudio y análisis de la estructura de la base de datos	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Identificar y verificar cada una de las entidades</li><li>• Verificar la cardinalidad de cada una de las entidades</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_7 Analizar y definir el estándar de programación***

Tarea	
<b>Numero tarea:</b> TI_7	<b>Número historia:</b> HT_5
<b>Nombre tarea:</b> Analizar y definir el estándar de programación	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 16/07/2020	<b>Fecha fin:</b> 17/07/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea establecer el estándar de programación bajo el cual se registrará el desarrollo del proyecto	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Verificar que el estándar sea entendible</li><li>• Verificar que el estándar de codificación cumpla con las expectativas requeridas</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_8 Estudiar del archivo main del proyecto***

Tarea	
<b>Numero tarea:</b> TI_8	<b>Número historia:</b> HT_6
<b>Nombre tarea:</b> Estudiar del archivo main del proyecto	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 20/07/2020	<b>Fecha fin:</b> 24/07/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea estudiar el archivo main del proyecto e identificar sus componentes	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Documentar la estructura del archivo main del proyecto</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_9 Estudiar el archivo de configuración y conexión a la base de datos***

Tarea	
<b>Numero tarea:</b> TI_9	<b>Número historia:</b> HT_6
<b>Nombre tarea:</b> Estudiar el archivo de configuración y conexión a la base de datos	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 27/07/2020	<b>Fecha fin:</b> 31/07/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea estudiar el archivo de configuración y conexión a la base de datos	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Documentar la estructura del archivo de configuración y conexión a la base de datos</li></ul>	

Fuente: Núñez, Gilberth, 2021

### TI\_10 Estudiar el archivo modelos

Tarea	
Numero tarea: TI_10	Número historia: HT_6
Nombre tarea: Estudiar el archivo modelos	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 2
Fecha inicio: 03/08/2020	Fecha fin: 07/08/2020
Programador responsable: Gilberth Núñez	
Descripción: Como desarrollador se desea estudiar el archivo modelos	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>Documentar la estructura del archivo de modelos</li></ul>	

Fuente: Núñez, Gilberth, 2021

### TI\_11 Estudiar el archivo serializadores

Tarea	
Numero tarea: TI_11	Número historia: HT_6
Nombre tarea: Estudiar el archivo serializadores	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 2
Fecha inicio: 10/08/2020	Fecha fin: 14/08/2020
Programador responsable: Gilberth Núñez	
Descripción: Como desarrollador se desea estudiar el archivo serializadores	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>Documentar la estructura del archivo de serializadores</li></ul>	

Fuente: Núñez, Gilberth, 2021

### TI\_12 Estudiar el archivo de declaración de los servicios REST

Tarea	
Numero tarea: TI_12	Número historia: HT_6
Nombre tarea: Estudiar el archivo de declaración de los servicios REST	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 2
Fecha inicio: 17/08/2020	Fecha fin: 21/08/2020
Programador responsable: Gilberth Núñez	
Descripción: Como desarrollador se desea estudiar el archivo de declaración de los servicios REST	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>Documentar la estructura del archivo de declaración de los servicios REST</li></ul>	

Fuente: Núñez, Gilberth, 2021

### TI\_13 Estudiar el archivo de servicios REST

Tarea	
Numero tarea: TI_13	Número historia: HT_6
Nombre tarea: Estudiar el archivo de servicios REST	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 5
Fecha inicio: 24/08/2020	Fecha fin: 11/09/2020
Programador responsable: Gilberth Núñez	
Descripción: Como desarrollador se desea estudiar el archivo de servicios REST	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>Documentar la estructura del archivo de servicios REST</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_14 Desarrollar el archivo main del proyecto***

<b>Tarea</b>	
<b>Numero tarea:</b> TI_14	<b>Número historia:</b> HT_7
<b>Nombre tarea:</b> Desarrollar el archivo main del proyecto	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 14/09/2020	<b>Fecha fin:</b> 18/09/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea codificar el archivo main del proyecto e identificar sus componentes	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Creación de las clases declaradas en la base de datos</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_15 Desarrollar el archivo de configuración y conexión a la base de datos***

<b>Tarea</b>	
<b>Numero tarea:</b> TI_15	<b>Número historia:</b> HT_7
<b>Nombre tarea:</b> Desarrollar el archivo de configuración y conexión a la base de datos	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 21/09/2020	<b>Fecha fin:</b> 25/09/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea codificar el archivo de configuración y conexión a la base de datos	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Conexión a la base de datos PostgreSQL</li><li>• Conexión a la base de datos MongoDB</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_16 Desarrollar el archivo modelos***

<b>Tarea</b>	
<b>Numero tarea:</b> TI_16	<b>Número historia:</b> HT_7
<b>Nombre tarea:</b> Desarrollar el archivo modelos	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 28/09/2020	<b>Fecha fin:</b> 02/10/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea codificar el archivo modelos	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Documentar la estructura del archivo de modelos</li></ul>	

Fuente: Núñez, Gilberth, 2021

### *TI\_17 Desarrollar el archivo serializadores*

Tarea	
<b>Numero tarea:</b> TI_17	<b>Número historia:</b> HT_7
<b>Nombre tarea:</b> Desarrollar el archivo serializadores	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 05/10/2020	<b>Fecha fin:</b> 09/10/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea codificar el archivo serializadores	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Generar los esquemas de cada clase declarada</li></ul>	

Fuente: Núñez, Gilberth, 2021

### *TI\_18 Desarrollar el archivo de declaración de los servicios REST*

Tarea	
<b>Numero tarea:</b> TI_18	<b>Número historia:</b> HT_7
<b>Nombre tarea:</b> Desarrollar el archivo de declaración de los servicios REST	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 12/10/2020	<b>Fecha fin:</b> 16/10/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea codificar el archivo de declaración de los servicios REST	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Verificar rutas establecidas para consumir los recursos</li></ul>	

Fuente: Núñez, Gilberth, 2021

### *TI\_19 Desarrollar el archivo de servicios REST*

Tarea	
<b>Numero tarea:</b> TI_19	<b>Número historia:</b> HT_7
<b>Nombre tarea:</b> Desarrollar el archivo de servicios REST	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 5
<b>Fecha inicio:</b> 19/10/2020	<b>Fecha fin:</b> 06/11/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea codificar el archivo de servicios REST	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Desarrollo del método POST</li><li>• Desarrollo del método GET</li><li>• Desarrollo del método GET by id</li><li>• Desarrollo del método PUT</li><li>• Desarrollo del método DELETE</li></ul>	

Fuente: Núñez, Gilberth, 2021



### ***TI\_20 Definir las reglas gramaticales***

Tarea	
<b>Numero tarea:</b> TI_20	<b>Número historia:</b> HT_8
<b>Nombre tarea:</b> Definir las reglas gramaticales	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 3
<b>Fecha inicio:</b> 09/11/2020	<b>Fecha fin:</b> 20/11/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea definir las reglas gramaticales	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Definir estructuras para clases y atributos</li><li>• El asistente de código funciona correctamente</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_21 Establecer las reglas gramaticales para bases de datos tipo PostgreSQL***

Tarea	
<b>Numero tarea:</b> TI_21	<b>Número historia:</b> HT_8
<b>Nombre tarea:</b> Establecer las reglas gramaticales para bases de datos tipo PostgreSQL	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 23/11/2020	<b>Fecha fin:</b> 27/11/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea establecer las reglas gramaticales para bases de datos tipo PostgreSQL	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Definir estructuras para clases y atributos para BD tipo PostgreSQL</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_22 Establecer las reglas gramaticales para bases de datos tipo MongoDB***

Tarea	
<b>Numero tarea:</b> TI_22	<b>Número historia:</b> HT_8
<b>Nombre tarea:</b> Establecer las reglas gramaticales para bases de datos tipo MongoDB	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 30/11/2020	<b>Fecha fin:</b> 04/12/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea establecer las reglas gramaticales para bases de datos tipo MongoDB	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Definir estructuras para clases y atributos para BD tipo MongoDB</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_23 Analizar el proceso de generación del metamodelo***

Tarea	
<b>Numero tarea:</b> TI_23	<b>Número historia:</b> HT_9
<b>Nombre tarea:</b> Analizar el proceso de generación del metamodelo	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 07/12/2020	<b>Fecha fin:</b> 07/12/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea analizar el proceso de generación del metamodelo	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Documentar el proceso de generación del metamodelo</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_24 Generar el metamodelo con la herramienta de Eclipse***

Tarea	
<b>Numero tarea:</b> TI_24	<b>Número historia:</b> HT_9
<b>Nombre tarea:</b> Generar el metamodelo con la herramienta de Eclipse	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 08/12/2020	<b>Fecha fin:</b> 08/12/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea generar el metamodelo con la herramienta de Eclipse	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• Verificar y organizar el metamodelo generado</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_25 Codificar el generador del archivo modelo***

Tarea	
<b>Numero tarea:</b> TI_25	<b>Número historia:</b> HT_10
<b>Nombre tarea:</b> Codificar el generador del archivo modelo	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 09/12/2020	<b>Fecha fin:</b> 15/12/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea codificar el generador del archivo modelo	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• El archivo de los modelos se genera de acuerdo con la gramática</li></ul>	

Fuente: Núñez, Gilberth, 2021

### ***TI\_26 Codificar el generador del archivo serializadores***

Tarea	
<b>Numero tarea:</b> TI_26	<b>Número historia:</b> HT_10
<b>Nombre tarea:</b> Codificar el generador del archivo serializadores	
<b>Tipo de tarea:</b> <b>Desarrollo/Corrección/Mejora/Otra</b>	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 16/12/2020	<b>Fecha fin:</b> 22/12/2020
<b>Programador responsable:</b> Gilberth Núñez	
<b>Descripción:</b> Como desarrollador se desea codificar el generador del archivo serializadores	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>• El archivo de los serializadores se genera de acuerdo con la gramática</li></ul>	

Fuente: Núñez, Gilberth, 2021

### **TI\_27 Codificar el generador del archivo de declaración de los servicios REST**

Tarea	
Numero tarea: TI_27	Número historia: HT_10
Nombre tarea: Codificar el generador del archivo de declaración de los servicios REST	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 2
Fecha inicio: 23/12/2020	Fecha fin: 29/12/2020
Programador responsable: Gilberth Núñez	
Descripción: Como desarrollador se desea codificar el generador del archivo de declaración de los servicios REST	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>El archivo de declaración de los servicios REST se genera de acuerdo con la gramática</li></ul>	

Fuente: Núñez, Gilberth, 2021

### **TI\_28 Codificar el generador del archivo de servicios REST**

Tarea	
Numero tarea: TI_28	Número historia: HT_10
Nombre tarea: Codificar el generador del archivo de servicios REST	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 3
Fecha inicio: 30/12/2020	Fecha fin: 12/01/2021
Programador responsable: Gilberth Núñez	
Descripción: Como desarrollador se desea codificar el generador del archivo de servicios REST	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>Los archivos de servicios REST se genera de acuerdo con la gramática</li></ul>	

Fuente: Núñez, Gilberth, 2021

### **TI\_29 Codificar el generador del archivo main y de configuración**

Tarea	
Numero tarea: TI_29	Número historia: HT_10
Nombre tarea: Codificar el generador del archivo main y de configuración	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 2
Fecha inicio: 13/01/2021	Fecha fin: 19/01/2021
Programador responsable: Gilberth Núñez	
Descripción: Como desarrollador se desea codificar el generador del archivo main y de configuración	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>Los archivos main y de configuración se genera de acuerdo con la gramática</li></ul>	

Fuente: Núñez, Gilberth, 2021

### **TI\_30 Uso de la gramática propuesta**

Tarea	
Numero tarea: TI_30	Número historia: HT_11
Nombre tarea: Uso de la gramática propuesta	
Tipo de tarea: Desarrollo/Corrección/Mejora/Otra	Puntos estimados: 1
Fecha inicio: 20/01/2021	Fecha fin: 22/01/2021
Programador responsable: Gilberth Núñez	
Descripción: Como desarrollador se desea definir la base de datos y las clases para generar un proyecto	
<b>Pruebas de aceptación</b>	
<ul style="list-style-type: none"><li>Generar los archivos del proyecto</li></ul>	

Fuente: Núñez, Gilberth, 2021

## Anexo D1: Tablas de Pruebas de Aceptación

### *PA\_1 Generar la documentación de la información encontrada*

Prueba de aceptación	
<b>Código:</b> PA_1	<b>Tarea de Ingeniería:</b> Recolectar y analizar la información
<b>Nombre:</b> Generar la documentación de la información encontrada	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 19/06/2020
<b>Descripción:</b> Realizar la documentación sobre los trabajos encontrados enfocados en el desarrollo de un DSL	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre el desarrollo de los DSL.</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Seleccionar los documentos relevantes</li><li>• Analizar los documentos seleccionados</li><li>• Documentar la información relevante para el desarrollo del DSL</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Documentación clara y completa sobre el desarrollo de un DSL</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_2 Generar la documentación sobre las herramientas encontradas**

Prueba de aceptación	
<b>Código:</b> PA_2	<b>Tarea de Ingeniería:</b> Analizar las herramientas
<b>Nombre:</b> Generar la documentación sobre las herramientas encontradas	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 26/06/2020
<b>Descripción:</b> Realizar la documentación sobre las herramientas encontradas para el desarrollo del DSL	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre las herramientas para el desarrollo de los DSL.</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Seleccionar las herramientas relevantes</li><li>• Analizar las herramientas seleccionados</li><li>• Documentar la información relevante sobre las herramientas seleccionadas para el desarrollo del DSL</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Documentación clara y completa sobre las herramientas para el desarrollo de un DSL</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_3 Realizar la instalación de las herramientas***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_3	<b>Tarea de Ingeniería:</b> Instalación del software
<b>Nombre:</b> Realizar la instalación de las herramientas	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 03/07/2020
<b>Descripción:</b> Realizar la instalación de las herramientas que fueron seleccionadas para el desarrollo del DSL	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre las herramientas para el desarrollo de los DSL</li><li>• Previamente tener los instaladores ejecutables de las herramientas seleccionadas</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Realizar el proceso de instalación de las herramientas</li><li>• Verificar que las herramientas fueron correctamente instaladas</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Herramientas instaladas y listas para el proceso de desarrollo</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_4 Verificar el funcionamiento de las herramientas instaladas***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_4	<b>Tarea de Ingeniería:</b> Instalación del software
<b>Nombre:</b> Verificar el funcionamiento de las herramientas instaladas	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 03/07/2020
<b>Descripción:</b> Realizar ejercicios en las herramientas instaladas para verificar su correcto funcionamiento y compatibilidad.	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre las herramientas instaladas</li><li>• Herramientas instaladas</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar las herramientas instaladas</li><li>• Realizar ejercicios para verificar su funcionamiento</li><li>• Invocar a las librerías seleccionadas y verificar su compatibilidad</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Correcto funcionamiento de las herramientas</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_5 Documentar la estructura encontrada en los diferentes proyectos***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_5	<b>Tarea de Ingeniería:</b> Analizar proyectos desarrollados en <i>Python</i>
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Documentar la estructura encontrada en los diferentes proyectos</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 08/07/2020
<b>Descripción:</b> Documentar la estructura encontrada en los diferentes proyectos	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre los proyectos desarrollados en <i>Python</i></li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Seleccionar los proyectos</li><li>• Analizar los proyectos</li><li>• Identificar la estructura definida para dicho proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Identificar una estructura en común para el desarrollo de proyectos con <i>Python</i></li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_6 Verificar que los proyectos encontrados trabajen con las herramientas seleccionadas***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_6	<b>Tarea de Ingeniería:</b> Analizar proyectos desarrollados en <i>Python</i>
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Verificar que los proyectos encontrados trabajen con las herramientas seleccionadas</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 08/07/2020
<b>Descripción:</b> Verificar que los proyectos encontrados trabajen con las herramientas seleccionadas para reducir la probabilidad de definir mal una estructura	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre los proyectos desarrollados en <i>Python</i></li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Seleccionar los proyectos</li><li>• Analizar los proyectos</li><li>• Identificar que herramientas declaradas en dichos proyectos coincidan con las seleccionadas para el desarrollo de este proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Las herramientas encontradas en los proyectos coinciden con las seleccionadas</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_7 Documentar la estructura definida para la generación del proyecto**

Prueba de aceptación	
<b>Código:</b> PA_7	<b>Tarea de Ingeniería:</b> Definir la estructura en la que se generará los proyectos
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Documentar la estructura definida para la generación del proyecto</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 10/07/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Documentar la estructura definida para la generación del proyecto</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre los proyectos desarrollados en <i>Python</i></li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Analizar la estructura del proyecto</li><li>• Documentar la estructura definida</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Documentación clara y completa sobre la estructura encontrada en los proyectos y la definición de la estructura</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021



**PA\_8 Verificar que los proyectos encontrados trabajen con las herramientas seleccionadas**

Prueba de aceptación	
<b>Código:</b> PA_8	<b>Tarea de Ingeniería:</b> Definir la estructura en la que se generará los proyectos
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Verificar que los proyectos encontrados trabajen con las herramientas seleccionadas</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 10/07/2020
<b>Descripción:</b> Verificar que los proyectos encontrados trabajen con las herramientas seleccionadas para reducir la probabilidad de definir mal una estructura	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre los proyectos desarrollados en <i>Python</i></li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Seleccionar los proyectos</li><li>• Analizar los proyectos</li><li>• Identificar que herramientas declaradas en dichos proyectos coincidan con las seleccionadas para el desarrollo de este proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Las herramientas encontradas en los proyectos coinciden con las seleccionadas</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_9 Identificar y verificar cada una de las entidades**

Prueba de aceptación	
<b>Código:</b> PA_9	<b>Tarea de Ingeniería:</b> Estudio y análisis de la base de datos
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Identificar y verificar cada una de las entidades</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 15/07/2020
<b>Descripción:</b> Se identifica y verifica que cada una de las entidades se encuentran en el diseño conceptual de la base de datos y sus respectivos atributos	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar el diseño de la base de datos</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Analizar las entidades</li><li>• Analizar cada uno de los atributos correspondientes a las entidades</li><li>• Identificar la relación y cardinalidad entre entidades</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se verifica que cada entidad está bien definida y no entra en conflicto de redundancia de información.</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_10 Verificar la cardinalidad de cada una de las entidades***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_10	<b>Tarea de Ingeniería:</b> Estudio y análisis de la base de datos
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Verificar la cardinalidad de cada una de las entidades</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 15/07/2020
<b>Descripción:</b> Como programador necesito establecer correctamente la cardinalidad entre las entidades.	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar el diseño de la base de datos</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Visualizar entidades y sus relaciones</li><li>• Analizar la correspondencia de las entidades</li><li>• Analizar la cardinalidad entre todas las entidades y sus relaciones</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se tiene todas las entidades están correctamente definidas con su nombre, atributos, relaciones y sus respectivas cardinalidades</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_11 Verificar que el estándar sea entendible***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_11	<b>Tarea de Ingeniería:</b> Analizar y definir el estándar de programación
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Verificar que el estándar sea entendible</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 17/07/2020
<b>Descripción:</b> El estándar que se utilizará en el desarrollo del sistema debe contener especificaciones exactas de cómo se implementará en la codificación	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Tener un estándar definido para programar</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Investigar tipos de Estándar</li><li>• Definir el estándar a ser desarrollado en nuestro sistema</li><li>• Verificar que la estándar codificación contenga reglas tales como: variables, constantes y clases</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Que el estándar contenga todas las cláusulas necesarias para el desarrollo del sistema.</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_12 Verificar que el estándar de codificación cumpla con las expectativas requeridas**

Prueba de aceptación	
<b>Código:</b> PA_12	<b>Tarea de Ingeniería:</b> Analizar y definir el estándar de programación
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Verificar que el estándar de codificación cumpla con las expectativas requeridas</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 17/07/2020
<b>Descripción:</b> El estándar deberá contener especificaciones claras de cómo crear: clases, objetos, métodos.	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• El estándar de codificación para el desarrollo del sistema debe estar especificado.</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Analizar el estándar que se ha especificado para el desarrollo del proyecto.</li><li>• Verificar que es un estándar fácil de seguir y que permitirá desarrollar código homogéneo y comprensible.</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Obtener un único estándar para la codificación que se realizara en el proyecto.</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_13 Documentar la estructura del archivo main del proyecto**

Prueba de aceptación	
<b>Código:</b> PA_13	<b>Tarea de Ingeniería:</b> Estudiar del archivo main del proyecto
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Documentar la estructura del archivo main del proyecto</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 24/07/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Documentar la estructura del archivo main del proyecto</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre el archivo main de proyectos desarrollados en <i>Python</i></li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Analizar la estructura del archivo</li><li>• Documentar la estructura</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Documentación clara y completa sobre el archivo main</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_14 Documentar la estructura del archivo de configuración y conexión a la base de datos**

Prueba de aceptación	
<b>Código:</b> PA_14	<b>Tarea de Ingeniería:</b> Estudiar el archivo de configuración y conexión a la base de datos
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Documentar la estructura del archivo de configuración y conexión a la base de datos</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 31/07/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Documentar la estructura del archivo de configuración y conexión a la base de datos</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre el archivo de configuración y conexión a la base de datos</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Analizar la estructura del archivo</li><li>• Documentar la estructura</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Documentación clara y completa sobre el archivo de configuración y conexión a la base de datos</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_15 Documentar la estructura del archivo de modelos**

Prueba de aceptación	
<b>Código:</b> PA_15	<b>Tarea de Ingeniería:</b> Estudiar el archivo de modelos
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Documentar la estructura del archivo de modelos</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 07/08/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Documentar la estructura del archivo de modelos</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre el archivo de modelos</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Analizar la estructura del archivo</li><li>• Documentar la estructura</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Documentación clara y completa sobre el archivo de modelos</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_16 Documentar la estructura del archivo de serializadores***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_16	<b>Tarea de Ingeniería:</b> Estudiar el archivo de serializadores
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Documentar la estructura del archivo de serializadores</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 14/08/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Documentar la estructura del archivo de serializadores</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre el archivo de serializadores</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Analizar la estructura del archivo</li><li>• Documentar la estructura</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Documentación clara y completa sobre el archivo de serializadores</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_17 Documentar la estructura del archivo de declaración de los servicios REST***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_17	<b>Tarea de Ingeniería:</b> Estudiar el archivo de declaración de los servicios REST
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Documentar la estructura del archivo de declaración de los servicios REST</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 21/08/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Documentar la estructura del archivo de declaración de los servicios REST</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre el archivo de declaración de los servicios REST</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Analizar la estructura del archivo</li><li>• Documentar la estructura</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Documentación clara y completa sobre el archivo de declaración de los servicios REST</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_18 Documentar la estructura del archivo de servicios REST***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_18	<b>Tarea de Ingeniería:</b> Estudiar el archivo de servicios REST
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Documentar la estructura del archivo de servicios REST</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 11/09/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Documentar la estructura del archivo de servicios REST</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar una investigación sobre el archivo de servicios REST</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Analizar la estructura del archivo</li><li>• Documentar la estructura</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Documentación clara y completa sobre el archivo de servicios REST</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_19 Creación de las clases declaradas en la base de datos***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_19	<b>Tarea de Ingeniería:</b> Desarrollar el archivo main del proyecto
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Creación de las clases declaradas en la base de datos</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 18/09/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Crear las clases declaradas en la base de datos</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar el archivo modelos en el cual se declaran las clases para la conexión en la base de datos</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Importar el archivo modelos</li><li>• Iniciar la aplicación</li><li>• Creación de la base de datos</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se realiza correctamente la creación de la base de datos según las clases declaradas en el archivo de modelos</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_20 Conexión a la base de datos***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_20	<b>Tarea de Ingeniería:</b> Desarrollar el archivo de configuración y conexión a la base de datos
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Conexión a la base de datos PostgreSQL</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 25/09/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Conexión a la base de datos PostgreSQL</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente instalar un gestor de base de datos y crear una base de datos a la cual se desea conectar</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Generar el archivo de conexión hacia una base de datos de tipo PostgreSQL</li><li>• Importar el archivo de configuración en el archivo main</li><li>• Ejecutar el proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se realiza correctamente la conexión con la base de datos</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_21 Verificar la cardinalidad de cada una de las entidades***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_21	<b>Tarea de Ingeniería:</b> Desarrollar el archivo de configuración y conexión a la base de datos
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Conexión a la base de datos MongoDB</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 25/09/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Conexión a la base de datos MongoDB</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente instalar un gestor de base de datos y crear una base de datos a la cual se desea conectar</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Generar el archivo de conexión hacia una base de datos de tipo MongoDB</li><li>• Importar el archivo de configuración en el archivo main</li><li>• Ejecutar el proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se realiza correctamente la conexión con la base de datos</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### **PA\_22 Verificar las clases declaradas**

Prueba de aceptación	
<b>Código:</b> PA_22	<b>Tarea de Ingeniería:</b> Desarrollar el archivo modelos
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Verificar las clases declaradas</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 02/10/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Verificar que las clases declaradas sean las establecidas en el diseño</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar el archivo modelos</li><li>• Previamente realizar el modelo lógico de la base de datos</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Comparar las clases declaradas en el archivo modelos con el modelo lógico de la base de datos</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Las clases declaradas coinciden con el modelo lógico de la base de datos con sus respectivos atributos y relaciones</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### **PA\_23 Generar las tablas a partir del proyecto**

Prueba de aceptación	
<b>Código:</b> PA_23	<b>Tarea de Ingeniería:</b> Desarrollar el archivo modelos
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Generar las tablas a partir del proyecto</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 02/10/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Generar las tablas a partir del proyecto</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente instalar un ORM que permita la comunicación de las clases del sistema con la base de datos</li><li>• Conexión con la base de datos</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se realiza correctamente la creación de las tablas en la base de datos</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021



**PA\_24 Generar los esquemas de cada clase declarada**

Prueba de aceptación	
<b>Código:</b> PA_24	<b>Tarea de Ingeniería:</b> Desarrollar el archivo serializadores
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Generar los esquemas de cada clase declarada</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 09/10/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Generar los esquemas de cada clase declarada</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente realizar el archivo de modelos</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Verificar que cada clase del archivo modelos tenga su esquema</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Cada clase posee su respectivo esquema</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_25 Verificar la cardinalidad de cada una de las entidades**

Prueba de aceptación	
<b>Código:</b> PA_25	<b>Tarea de Ingeniería:</b> Desarrollar el archivo de declaración de los servicios REST
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Verificar rutas establecidas para consumir los recursos</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 16/10/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Verificar rutas establecidas para consumir los recursos</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente instalar el software Postman</li><li>• Previamente desarrollar el archivo modelos</li><li>• Previamente desarrollar el archivo de serializadores</li><li>• Previamente desarrollar los métodos REST</li><li>• Previamente declarar la ruta por la que se accederá a los servicios</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Ejecutar el software Postman</li><li>• Seleccionar el método que será testeado en este caso el GET</li><li>• Ingresar la dirección para consumir el servicio</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se consume los servicios</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### **PA\_26 Uso del método POST**

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_26	<b>Tarea de Ingeniería:</b> Desarrollar el archivo de servicios REST
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Uso del método POST</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 06/11/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Uso del método POST</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente instalar el software Postman</li><li>• Previamente desarrollar el archivo modelos</li><li>• Previamente desarrollar el archivo de serializadores</li><li>• Previamente desarrollar los métodos REST</li><li>• Previamente declarar la ruta por la que se accederá a los servicios</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Ejecutar el software Postman</li><li>• Seleccionar el método que será testeado en este caso el POST</li><li>• Ingresar la dirección para consumir el servicio</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se consume correctamente el servicio</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

**Fuente:** Núñez, Gilberth, 2021

### PA\_27 Uso del método GET

Prueba de aceptación	
<b>Código:</b> PA_27	<b>Tarea de Ingeniería:</b> Desarrollar el archivo de servicios REST
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Uso del método GET</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 06/11/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Uso del método GET</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente instalar el software Postman</li><li>• Previamente desarrollar el archivo modelos</li><li>• Previamente desarrollar el archivo de serializadores</li><li>• Previamente desarrollar los métodos REST</li><li>• Previamente declarar la ruta por la que se accederá a los servicios</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Ejecutar el software Postman</li><li>• Seleccionar el método que será testeado en este caso el GET</li><li>• Ingresar la dirección para consumir el servicio</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se consume correctamente el servicio</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### PA\_28 Uso del método GET by id

Prueba de aceptación	
<b>Código:</b> PA_28	<b>Tarea de Ingeniería:</b> Desarrollar el archivo de servicios REST
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Uso del método GET by id</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 06/11/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Uso del método GET by id</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente instalar el software Postman</li><li>• Previamente desarrollar el archivo modelos</li><li>• Previamente desarrollar el archivo de serializadores</li><li>• Previamente desarrollar los métodos REST</li><li>• Previamente declarar la ruta por la que se accederá a los servicios</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Ejecutar el software Postman</li><li>• Seleccionar el método que será testeado en este caso el GET by id</li><li>• Ingresar la dirección para consumir el servicio</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se consume correctamente el servicio</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### PA\_29 Uso del método PUT

Prueba de aceptación	
<b>Código:</b> PA_29	<b>Tarea de Ingeniería:</b> Desarrollar el archivo de servicios REST
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Uso del método PUT</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 06/11/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Uso del método PUT</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente instalar el software Postman</li><li>• Previamente desarrollar el archivo modelos</li><li>• Previamente desarrollar el archivo de serializadores</li><li>• Previamente desarrollar los métodos REST</li><li>• Previamente declarar la ruta por la que se accederá a los servicios</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Ejecutar el software Postman</li><li>• Seleccionar el método que será testeado en este caso el PUT</li><li>• Ingresar la dirección para consumir el servicio</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se consume correctamente el servicio</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### PA\_30 Uso del método DELETE

Prueba de aceptación	
<b>Código:</b> PA_30	<b>Tarea de Ingeniería:</b> Desarrollar el archivo de servicios REST
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Uso del método DELETE</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 06/11/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Uso del método DELETE</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Previamente instalar el software Postman</li><li>• Previamente desarrollar el archivo modelos</li><li>• Previamente desarrollar el archivo de serializadores</li><li>• Previamente desarrollar los métodos REST</li><li>• Previamente declarar la ruta por la que se accederá a los servicios</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Ejecutar el software Postman</li><li>• Seleccionar el método que será testeado en este caso el DELETE</li><li>• Ingresar la dirección para consumir el servicio</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se consume correctamente el servicio</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### PA\_31 Definir estructuras para clases y atributos

Prueba de aceptación	
<b>Código:</b> PA_31	<b>Tarea de Ingeniería:</b> Definir las reglas gramaticales
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Definir estructuras para clases y atributos</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 20/11/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Definir estructuras para clases y atributos</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• No hay conflicto el momento de ejecutar una instancia con la gramática</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_32 El asistente de código funciona correctamente***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_32	<b>Tarea de Ingeniería:</b> Definir las reglas gramaticales
<ul style="list-style-type: none"><li>• <b>Nombre:</b> El asistente de código funciona correctamente</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 20/11/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> El asistente de código funciona correctamente según lo definido por las reglas gramaticales y no entra en conflicto con otra regla</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Ejecutar una instancia de la gramática del proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• El asistente de código funciona correctamente</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_33 Definir estructuras para clases y atributos para BD tipo PostgreSQL***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_33	<b>Tarea de Ingeniería:</b> Definir las reglas gramaticales
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Definir estructuras para clases y atributos para BD tipo PostgreSQL</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 27/11/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Definir estructuras para clases y atributos para BD tipo PostgreSQL</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• No hay conflicto el momento de ejecutar una instancia con la gramática</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021



### ***PA\_34 Definir estructuras para clases y atributos para BD tipo MongoDB***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_34	<b>Tarea de Ingeniería:</b> Definir las reglas gramaticales
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Definir estructuras para clases y atributos para BD tipo MongoDB</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 04/12/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Definir estructuras para clases y atributos para BD tipo MongoDB</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• No hay conflicto el momento de ejecutar una instancia con la gramática</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_35 Documentar el proceso de generación del metamodelo***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_35	<b>Tarea de Ingeniería:</b> Analizar el proceso de generación del metamodelo
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Documentar el proceso de generación del metamodelo</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 07/12/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Documentar el proceso de generación del metamodelo</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Analizar las herramientas seleccionados</li><li>• Documentar el proceso de generación del metamodelo de la gramática</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Documentación clara sobre el proceso de generación del metamodelo</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_36 Verificar y organizar el metamodelo generado***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_36	<b>Tarea de Ingeniería:</b> Generar el metamodelo con la herramienta de Eclipse
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Verificar y organizar el metamodelo generado</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 08/12/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Verificar y organizar el metamodelo generado</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li><li>• Generar el metamodelo de la gramática</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Generar el metamodelo</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• El metamodelo es generado correctamente según la gramática definida y se encuentra organizado</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

### ***PA\_37 El archivo de los modelos se genera de acuerdo con la gramática***

<b>Prueba de aceptación</b>	
<b>Código:</b> PA_37	<b>Tarea de Ingeniería:</b> Codificar el generador del archivo modelo
<ul style="list-style-type: none"><li>• <b>Nombre:</b> El archivo de los modelos se genera de acuerdo con la gramática</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 15/12/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> El archivo de los modelos se genera de acuerdo con la gramática propuesta</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Declarar las clases con el uso de la gramática propuesta</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se genera el archivo modelos con sus clases y atributos</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_38 El archivo de los serializadores se genera de acuerdo con la gramática**

Prueba de aceptación	
<b>Código:</b> PA_38	<b>Tarea de Ingeniería:</b> Codificar el generador del archivo serializadores
<ul style="list-style-type: none"><li>• <b>Nombre:</b> El archivo de los serializadores se genera de acuerdo con la gramática</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 22/12/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> El archivo de los serializadores se genera de acuerdo con la gramática propuesta</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Declarar las clases con el uso de la gramática propuesta</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se genera el archivo serializadores con sus clases y atributos</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_39 El archivo de declaración de los servicios REST se genera de acuerdo con la gramática**

Prueba de aceptación	
<b>Código:</b> PA_39	<b>Tarea de Ingeniería:</b> Codificar el generador del archivo serializadores
<ul style="list-style-type: none"><li>• <b>Nombre:</b> El archivo de declaración de los servicios REST se genera de acuerdo con la gramática</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 29/12/2020
<ul style="list-style-type: none"><li>• <b>Descripción:</b> El archivo de declaración de los servicios REST se genera de acuerdo con la gramática propuesta</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Declarar las clases con el uso de la gramática propuesta</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se genera el archivo de declaración de los servicios REST con sus clases y métodos</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_40 Los archivos de servicios REST se genera de acuerdo con la gramática**

Prueba de aceptación	
<b>Código:</b> PA_40	<b>Tarea de Ingeniería:</b> Codificar el generador del archivo de servicios REST
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Los archivos de servicios REST se genera de acuerdo con la gramática</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 12/01/2021
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Los archivos de servicios REST se genera de acuerdo con la gramática propuesta</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Declarar las clases con el uso de la gramática propuesta</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se genera los archivos de servicios REST con sus clases y métodos</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_41 Los archivos main y de configuración se genera de acuerdo con la gramática**

Prueba de aceptación	
<b>Código:</b> PA_41	<b>Tarea de Ingeniería:</b> Codificar el generador del archivo main y de configuración
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Los archivos main y de configuración se genera de acuerdo con la gramática</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 19/01/2021
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Los archivos main y de configuración se genera de acuerdo con la gramática propuesta</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li><li>• Declarar las clases con el uso de la gramática propuesta</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se genera los archivos main y de configuración correctamente</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

Fuente: Núñez, Gilberth, 2021

**PA\_42 Generar los archivos del proyecto**

Prueba de aceptación	
<b>Código:</b> PA_42	<b>Tarea de Ingeniería:</b> Uso de la gramática propuesta
<ul style="list-style-type: none"><li>• <b>Nombre:</b> Generar los archivos del proyecto</li></ul>	
<b>Responsable:</b> Gilberth Núñez	<b>Fecha:</b> 22/01/2021
<ul style="list-style-type: none"><li>• <b>Descripción:</b> Se generará los archivos necesarios para levantar el proyecto en cualquier otro IDE sin problema</li></ul>	
<b>Condiciones de Ejecución</b> <ul style="list-style-type: none"><li>• Instalar el IDE Eclipse</li><li>• Instalar el framework Xtext</li></ul>	
<b>Pasos de Ejecución</b> <ul style="list-style-type: none"><li>• Ejecutar el proyecto</li></ul>	
<b>Resultado Esperado</b> <ul style="list-style-type: none"><li>• Se generan los archivos correctamente</li></ul>	
<b>Evaluación de la prueba:</b> Exitosa	

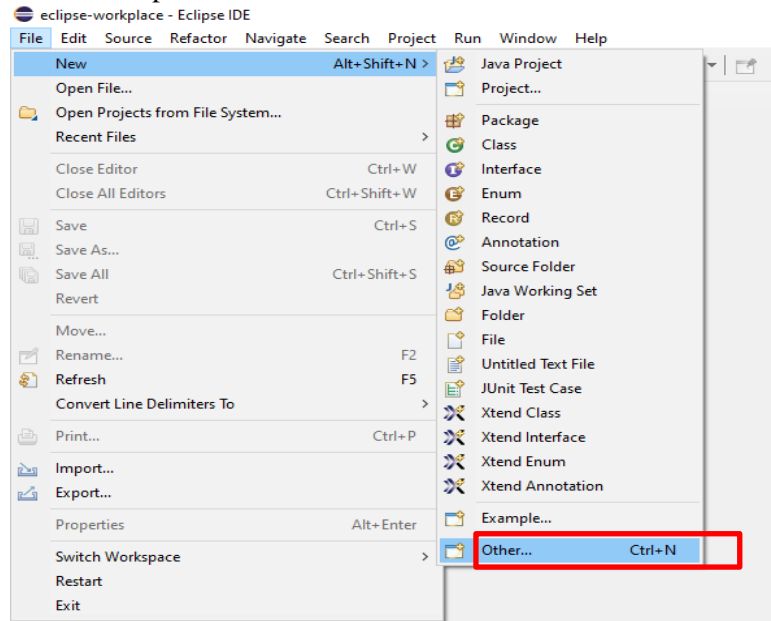
Fuente: Núñez, Gilberth, 2021

## Anexo E: Manual de Usuario

### Despliegue del DSL

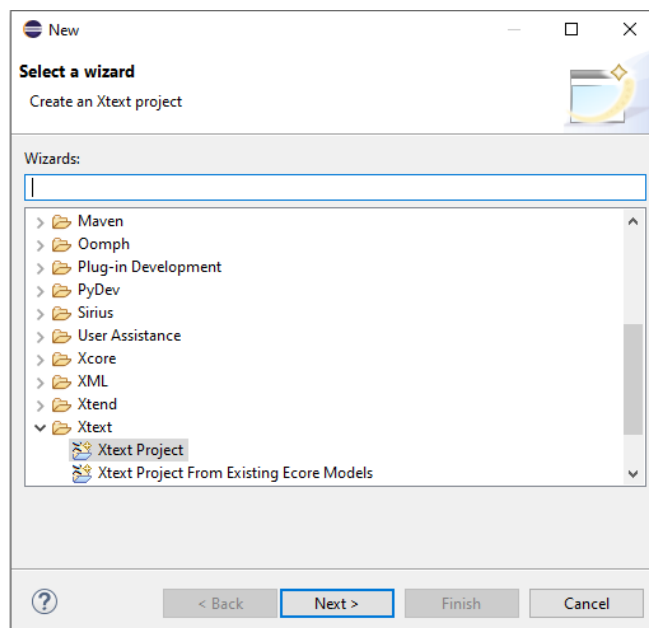
Para el despliegue del sistema se necesita el IDE Eclipse e instalar el framework xtext y el compilador xtend.

Se debe crear un nuevo proyecto xtext para ello hay q seleccionar del menú la opción archivo, nuevo y seleccionamos la opción de otro.



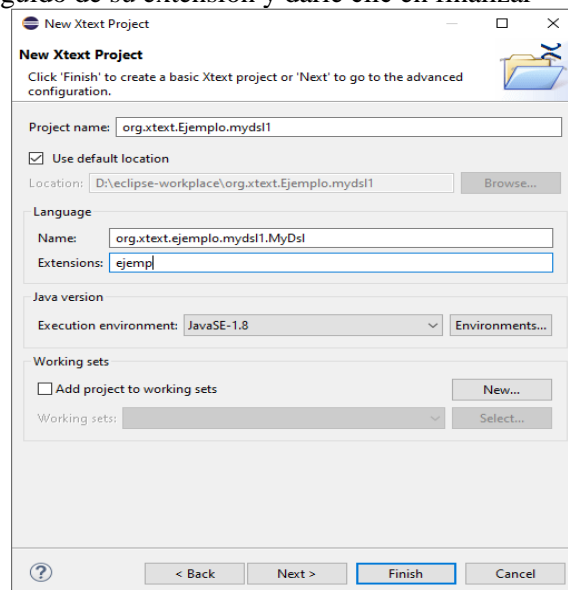
*Figura 3: Creación de un nuevo proyecto*

Aparece una ventana en la cual debemos seleccionar la carpeta xtext y seleccionar la opción de proyecto Xtext.



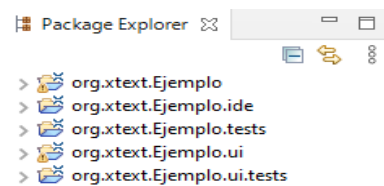
*Figura 4: Selección de un proyecto xtext*

Aparece una nueva ventana en la cual debemos ingresar el nombre del proyecto, después el nombre del lenguaje seguido de su extensión y darle clic en finalizar



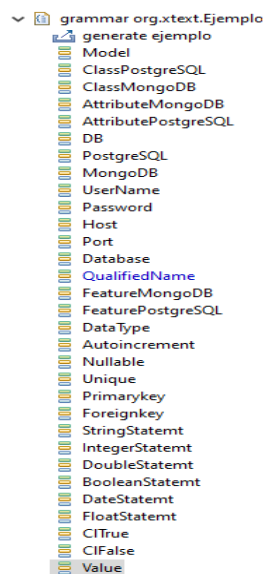
**Figura 5: Nuevo proyecto xtext**

Una vez creado el proyecto nos aparecen los siguientes archivos



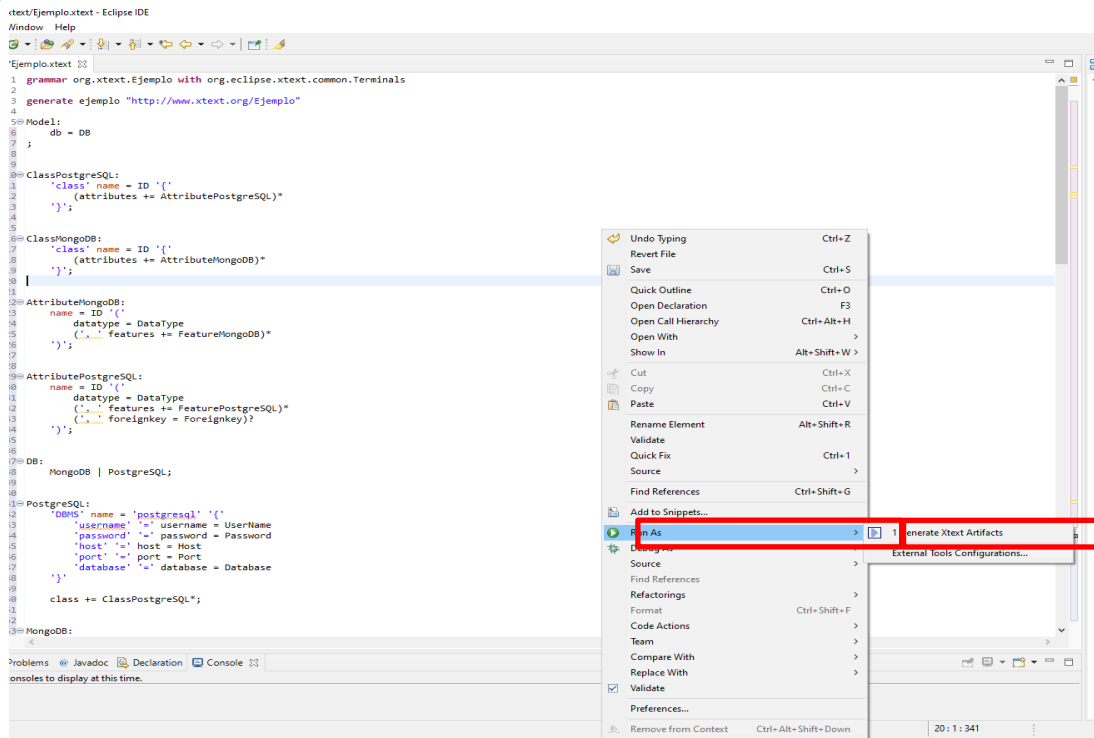
**Figura 6: Archivos de un proyecto xtext**

De manera automática se abre por defecto el archivo xtext donde se colocará la gramática de nuestro DSL, de tal forma que nos mostrará la estructura en el panel derecho de la siguiente forma



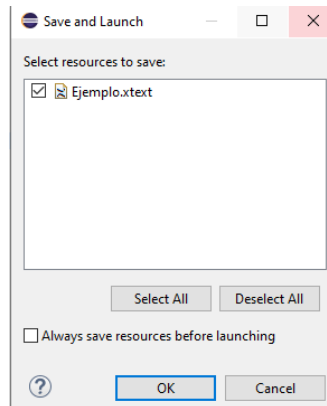
**Figura 7: Gramática del DSL**

Una vez que se tiene la gramática en el archivo xtext debemos dar un clic derecho sobre el área de trabajo, seleccionamos la opción de ejecutar como y después seleccionamos la opción de generar artefactos xtext.



**Figura 8: Generación de artefactos xtext**

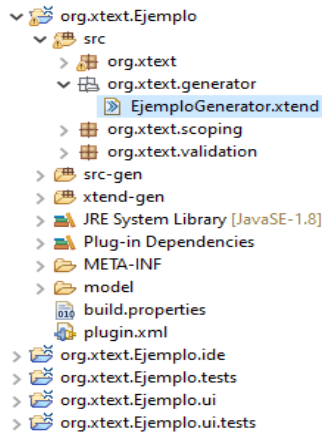
Nos aparece una nueva ventana donde seleccionamos nuestro archivo de gramática y damos clic en el botón ok.



**Figura 9: Selección del archivo al cual del cual se generará los artefactos**

Una vez generado los archivos debemos ubicarnos en el archivo xtend que fue generado.

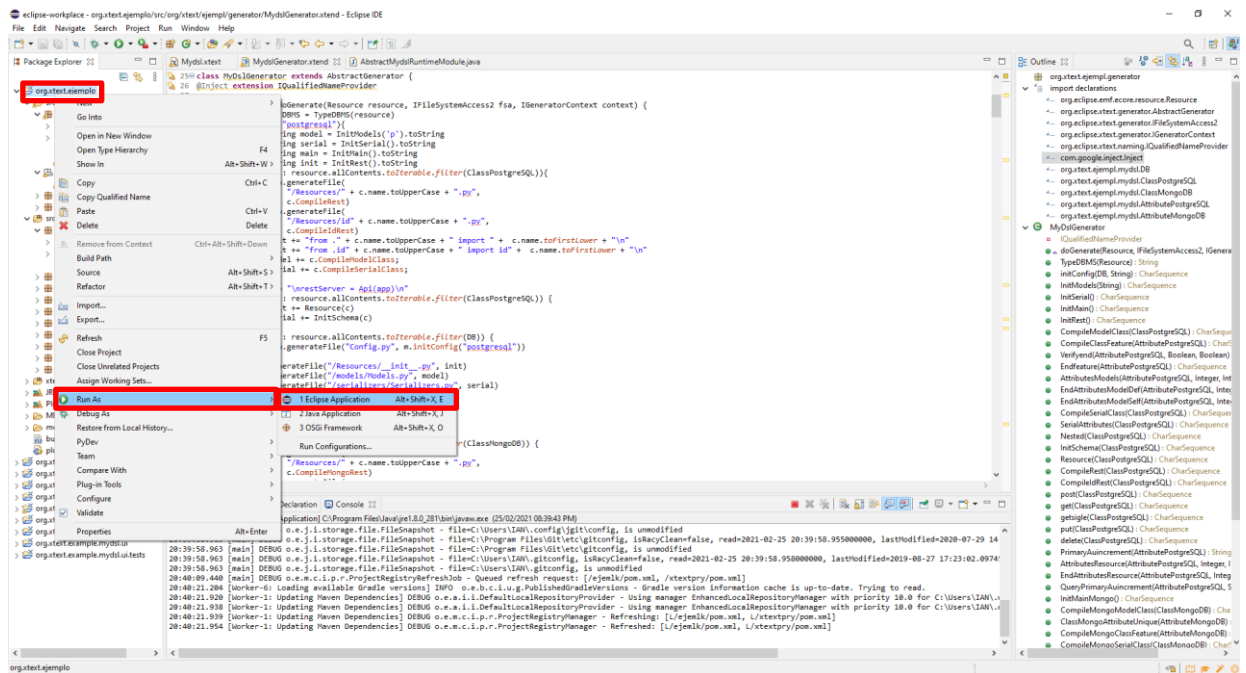




**Figura 10: Ubicación del archivo xtend**

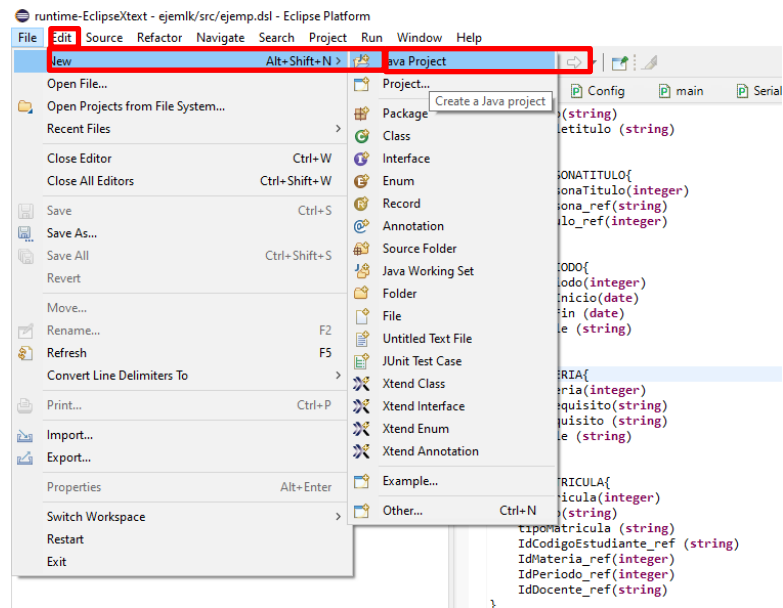
En este archivo debemos pegar el generador de código xtend para posteriormente proceder a ejecutar una instancia del proyecto

Para ejecutar una instancia del proyecto nos ubicamos en el proyecto damos clic derecho nos ubicamos en ejecutar como y seleccionamos como aplicación Eclipse



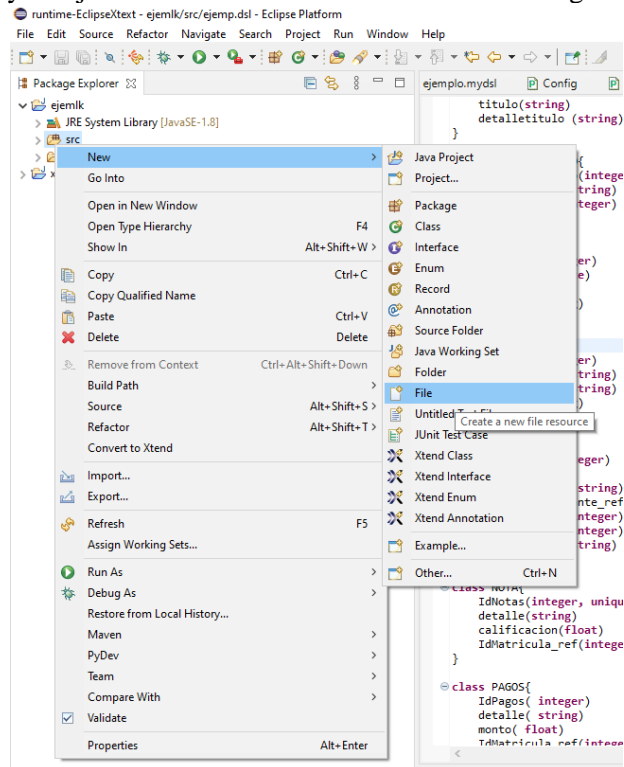
**Figura 11: Ejecutar una instancia del proyecto**

Se abre una nueva pestaña del IDE Eclipse, aquí debemos crear un proyecto ubicándonos en la pestaña de archivo, nos ubicamos en la opción nuevo y se desplegará un submenú en el cual seleccionaremos la opción proyecto java



**Figura 12: Creación de un proyecto java**

Una vez creado el proyecto java creamos un archivo a el cual se le asignara el formato del DSL



**Figura 13: Creación de un archivo**

Aparecerá un aviso para transformar el tipo del archivo en formato del DSL el cual se aceptará y se podrá utilizar la gramática para generar un proyecto Python.

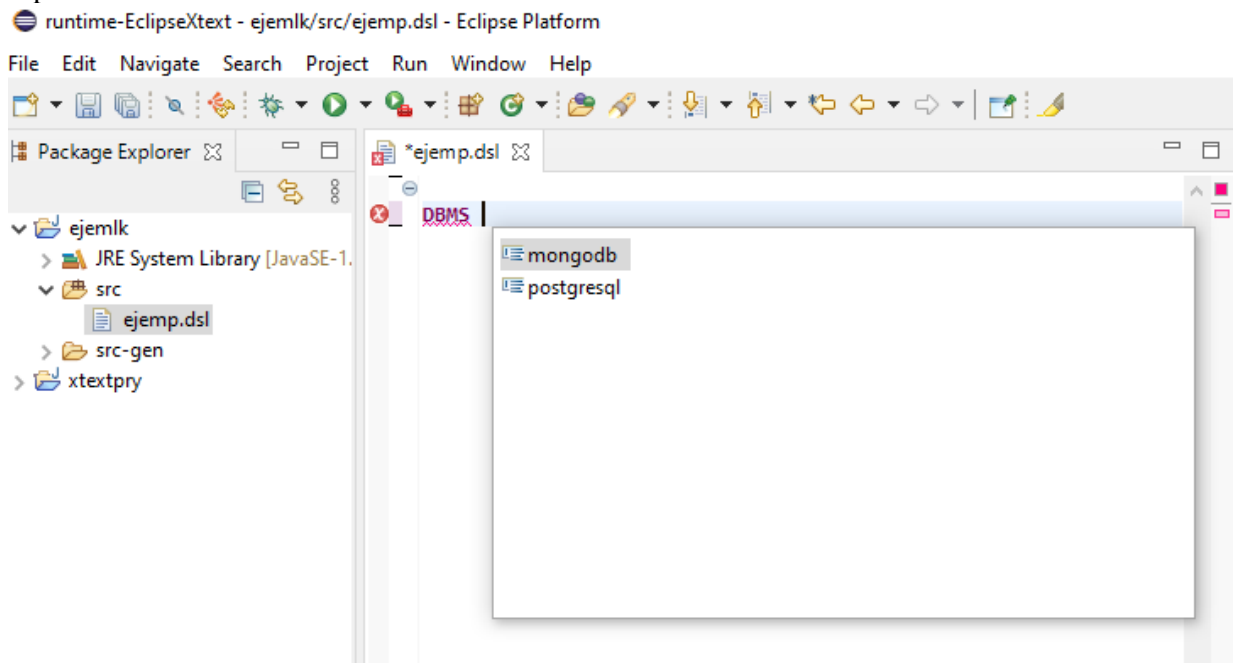
## Gramática del DSL

Para el desarrollo de la gramática se planteó dos escenarios: un proyecto con conexión a una base de datos relacional y otro proyecto con conexión a una base de datos no relacional.

La estructura que comparten ambos proyectos en su gramática es que ambos inician con la declaración de la base de datos y posterior a ellos se deben definir las clases con sus respectivos atributos, a continuación, se detalla la estructura y atributos que pueden ser utilizados en el proyecto.

- Conexión a una base de datos relacional

Se debe declarar la base de datos, se puede utilizar el asistente de código del IDE para apoyarnos con el uso de la gramática, para activar el asistente de código debemos usar las teclas `ctrl + barra espaciadora`



*Figura 14: Uso del asistente de código*

Para la declaración de la base de datos se debe iniciar con las siglas DBMS a continuación se declara el tipo de base de datos y las credenciales con las cuales se conectará a la base de datos.

```
DBMS postgresql{  
    username=postgres  
    password=postgres  
    host=localhost  
    port=5432  
    database=Prueba  
}
```

*Figura 15: Ejemplo de la declaración de una clase y las credenciales para conectarse a una base de datos tipo PostgreSQL*

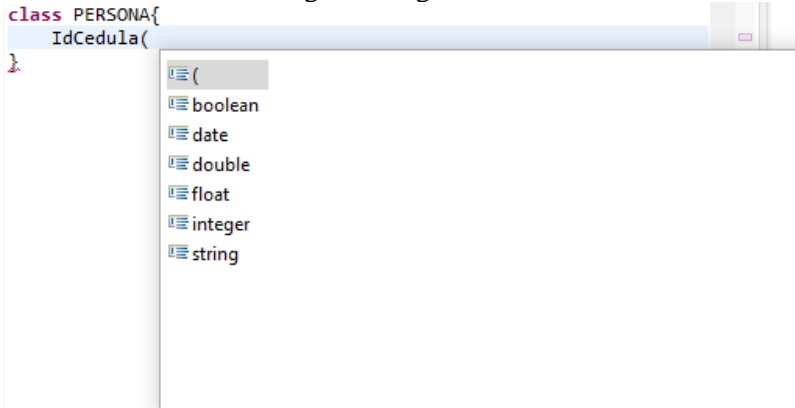
Una vez establecida la base de datos se procede a la generación de las clases, para declarar un clase se debe anteponer la palabra reservada **class**, a continuación, el nombre de la clase seguido de llaves.

```
class PERSONA{
}

```

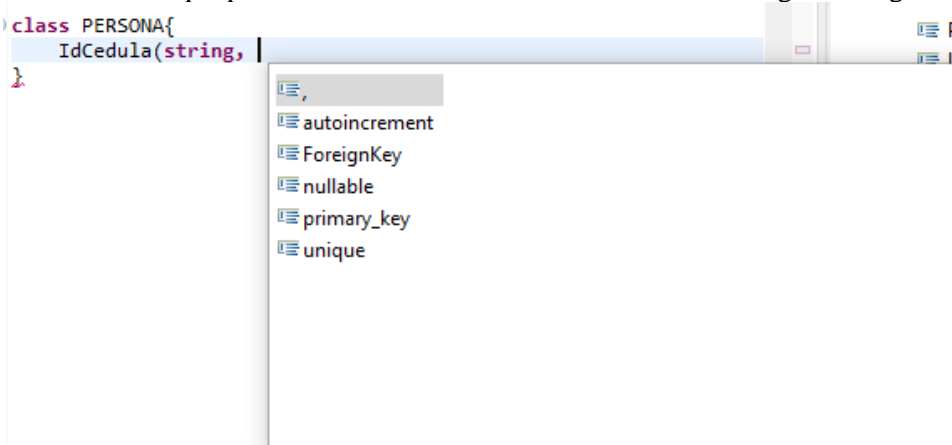
**Figura 16: Declaración de una clase**

Entre las llaves se deben declarar los atributos que pertenecen a la clase para ello, primero se debe declarar el nombre del atributo seguido de paréntesis, entre los paréntesis se debe declarar el tipo de dato y las características en caso de ser necesarias, los tipos de datos que pueden ser utilizados son los mostrados en la siguiente figura



**Figura 17: Tipos de datos que puede ser usada en proyectos PostgreSQL con el DSL**

Después de declarar el tipo de datos se puede agregar características seguidas de una coma, las características que pueden ser utilizadas son las mostradas en la siguiente figura



**Figura 18: Características que pueden ser usadas en proyectos PostgreSQL con el DSL**

Para activar las características se debe usar las palabras reservadas **true** o **false** seguidos del signo igual (=)

```
class PERSONA{
  IdCedula (string, primary_key=True)
}

```

**Figura 19: Ejemplo de uso de la característica de un proyecto conectado a una base de datos relacional**

Para el uso de la característica clave foránea, esta debe ser especificada antes de cerrar los paréntesis del atributo como se muestra en el siguiente fragmento de código

```
class PERSONATITULO{
    idPersonaTitulo(integer, autoincrement=True, primary_key=True, nullable=False)
    IdTitulo_ref(integer, nullable=False, ForeignKey())
}
```

**Figura 20: Declaración de la característica clave foránea**

Dentro de los paréntesis de la característica clave foránea se debe declarar la clase seguido del signo igual (=) y finalmente la clase seguido de un punto seguido de la característica a la cual se hará referencia, por ejemplo

```
ForeignKey(PERSONA = PERSONA.IdCedula)
```

**Figura 21: Ejemplo de uso de la característica clave foránea en el DSL**

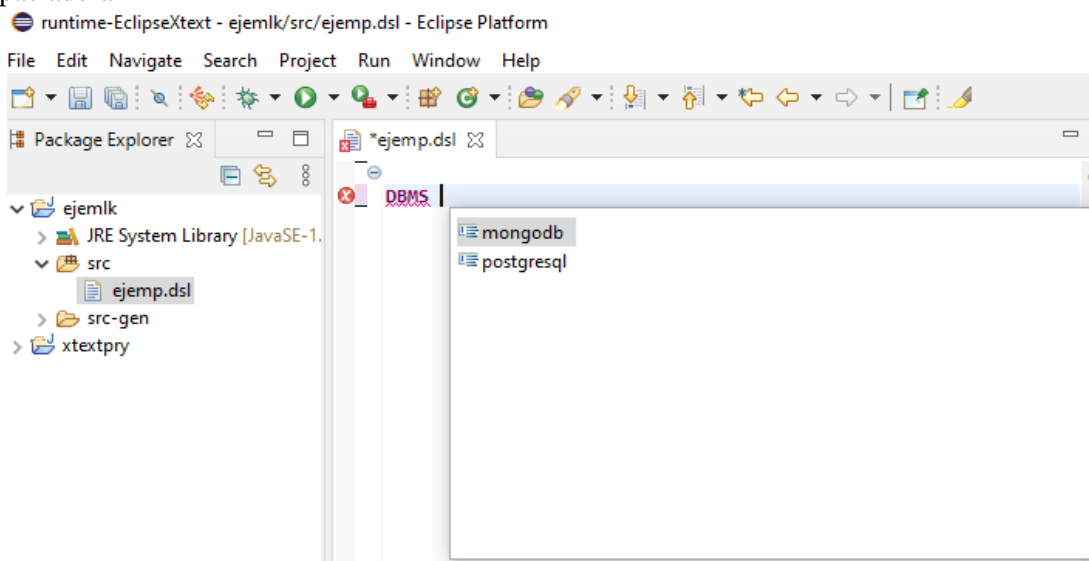
Un fragmento de una clase perteneciente a un proyecto con conexión a una base de datos relacional se muestra en la siguiente figura

```
class PERSONATITULO{
    idPersonaTitulo(integer, autoincrement=True, primary_key=True, nullable=False)
    IdPersona_ref(string, ForeignKey(PERSONA = PERSONA.IdCedula))
    IdTitulo_ref(integer, nullable=False, ForeignKey(TITULO= TITULO.IdTitulo))
}
```

**Figura 22: Ejemplo de una clase de un proyecto con conexión a una base de datos no relacional**

- Conexión a una base de datos no relacional

Se debe declarar la base de datos, se puede utilizar el asistente de código del IDE para apoyarnos con el uso de la gramática, para activar el asistente de código debemos usar las teclas ctrl + barra espaciadora



**Figura 23: Uso del asistente de código**

Para la declaración de la base de datos se debe iniciar con las siglas DBMS a continuación se declara el tipo de base de datos y las credenciales con las cuales se conectará a la base de datos.

```

DBMS mongodb{
  username=mongo
  password=sistemas
  host=localhost
  port=5432
  database=Prueba
}

```

**Figura 24: Ejemplo de la declaración de una clase y las credenciales para conectarse a una base de datos tipo MongoDB**

Una vez establecida la base de datos se procede a la generación de las clases, para declarar un clase se debe anteponer la palabra reservada **class**, a continuación, el nombre de la clase seguido de llaves.

```

class PERSONA{
}

```

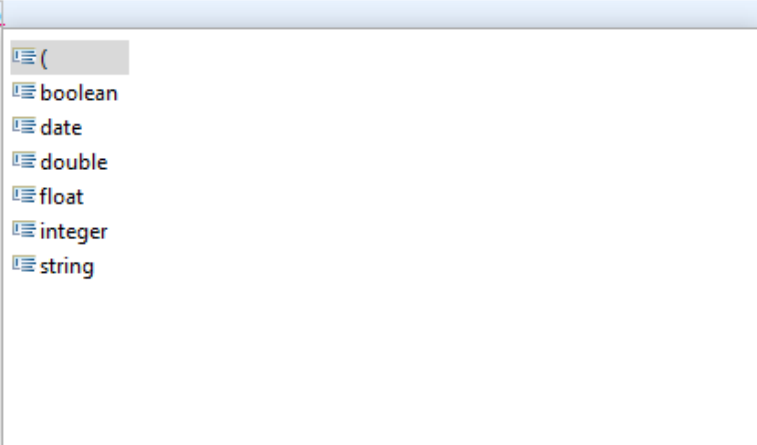
**Figura 25: Declaración de una clase**

Entre las llaves se deben declarar los atributos que perteneces a la clase para ello, primero se debe declarar el nombre del atributo seguido de paréntesis y entre los paréntesis se debe declarar el tipo de dato y la característica en caso de ser necesaria, los tipos de datos que pueden ser utilizados son los mostrados en la siguiente figura

```

class PERSONA{
  IdCedula ( )
}

```



The screenshot shows a code editor with the following code: `class PERSONA{`, `IdCedula ( )`, and `}`. A dropdown menu is open below the opening parenthesis of `IdCedula ( )`, listing the following data types: `boolean`, `date`, `double`, `float`, `integer`, and `string`.

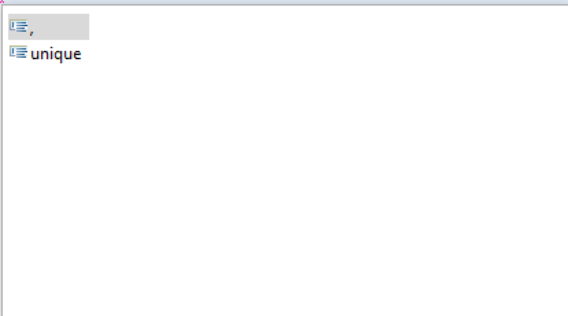
**Figura 26: Tipos de datos que puede ser usada en proyectos MongoDB con el DSL**

Después de declarar el tipo de datos se puede agregar una característica seguida de una coma, la característica que puede ser utilizada es la mostrada en la siguiente figura

```

class PERSONA{
  IdCedula (string, )
}

```



The screenshot shows a code editor with the following code: `class PERSONA{`, `IdCedula (string, )`, and `}`. A dropdown menu is open below the comma in `IdCedula (string, )`, listing the following characteristic: `unique`.

**Figura 27: Característica que puede ser usada en proyectos MongoDB con el DSL**

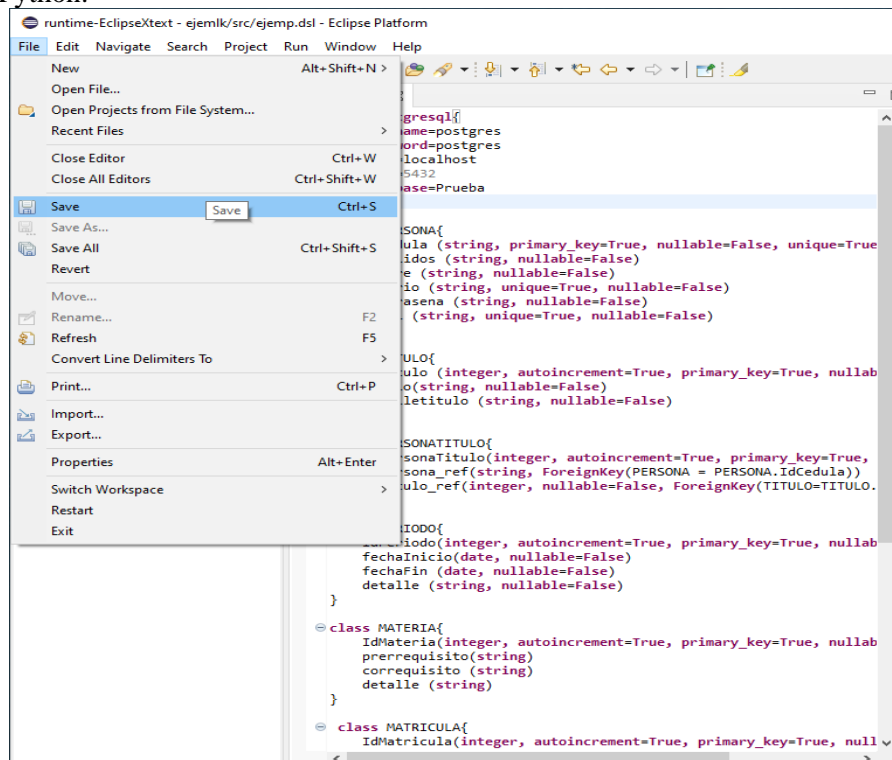
Para activar la característica se debe usar las palabras reservadas **true** o **false** seguidos del signo igual (=)

```
class PERSONA{
    IdCedula (string, unique=True)
}
```

*Figura 28: ejemplo de una clase de un proyecto con conexión a una base de datos no relacional*

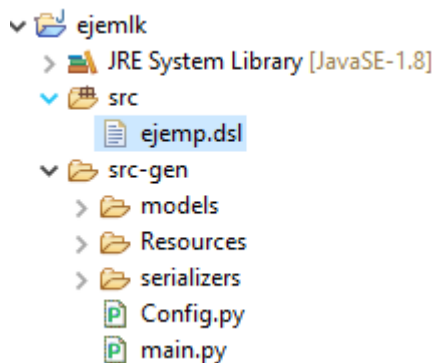
### Anexo E3: Generación e implementación del proyecto

Una vez declaradas las clases del proyecto se debe guardar los cambios y se generara el proyecto Python.



*Figura 29: Opción guardar cambios para generar el proyecto Python*

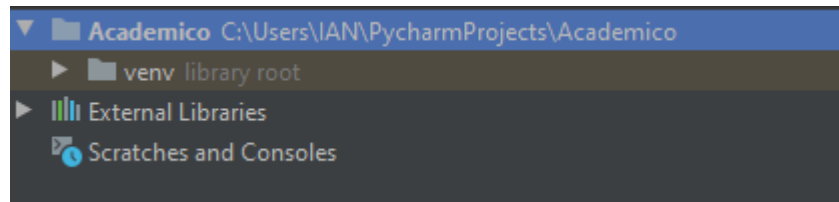
El proyecto se puede visualizar en el explorador de paquetes, en el cual se creará bajo una carpeta denominada **src-gen** donde se puede observar los archivos que fueron generados



*Figura 30. Archivo formato DSL creado*

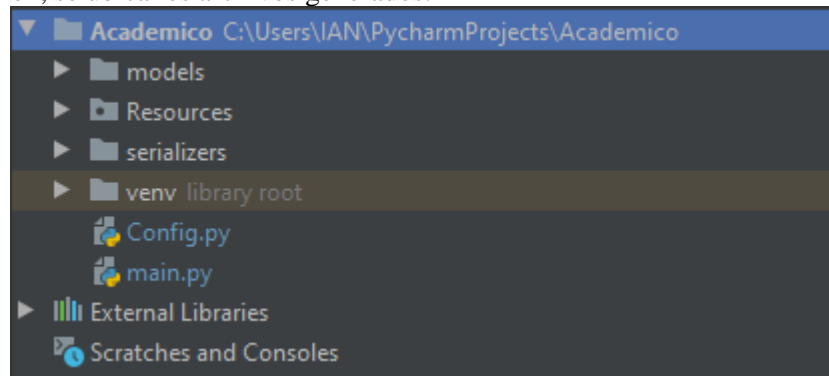
Para abrir ejecutar el proyecto se recomienda abrir el archivo denominado **requirements.txt** en el cual se encuentran las librerías que deben ser instaladas con sus versiones.

El contenido de la carpeta generada (**src-gen**) debe ser ubicada en el entorno donde se desea ejecutar, para ello se creó un proyecto en el IDE Pycharm llamado Académico con su entorno virtual.



**Figura 31: Estructura de un proyecto Python**

A continuación, se ubica los archivos generados.



**Figura 32: Archivos generados ubicados en el proyecto Python**

Y procedemos a ejecutar el proyecto

```
* Serving Flask app "main" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 940-427-349
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

**Figura 33: Ejecución del proyecto mostrado en la consola**

Finalmente se observa que el proyecto fue ejecutado correctamente.





ESCUELA SUPERIOR POLITÉCNICA DE  
CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL  
APRENDIZAJE



UNIDAD DE PROCESOS TÉCNICOS  
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 20/05/2021

**INFORMACIÓN DEL AUTOR**

**Nombres – Apellidos:** GILBERTH IANROLANDO NÚÑEZ RIBADENEYRA

**INFORMACIÓN INSTITUCIONAL**

**Facultad:** INFORMÁTICA Y ELECTRÓNICA

**Carrera:** INGENIERÍA EN SISTEMAS

**Título a optar:** INGENIERO EN SISTEMAS INFORMÁTICOS

**f. Analista de Biblioteca responsable:**



Firmado electrónicamente por:  
ELIZABETH  
FERNANDA AREVALO  
MEDINA

