

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

Certamen 1 ADOO

Integrantes: Vania Castro
Thomas Cordova
Atilio Tudesca
Luis Vega

Asignatura: Análisis y diseño orientado a objetos.

Paralelo: 301.

Fecha de entrega: 30-09-2021.

Docente: Jocelyn Gonzales.

Desarrollo

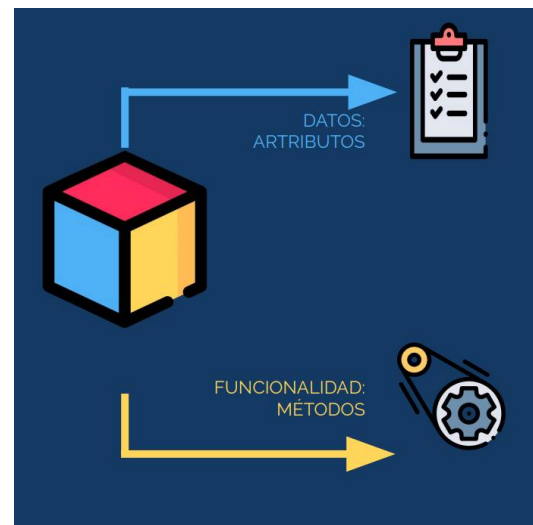
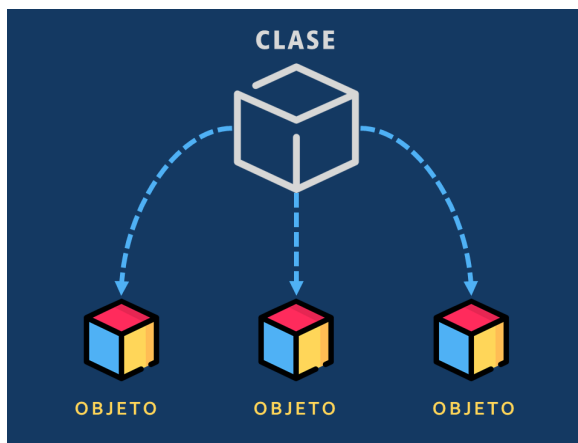
1.- ¿Qué es la Orientación a Objeto?

Para definir qué es la orientación a objeto, primero que debemos saber es que es un objeto, un objeto es considerado un campo de datos, de los cuales cada uno de estos poseen nombre, datos y funcionalidades, que en la programación orientado a objetos son denominados atributos y métodos.

Los atributos son las características de los objetos, pueden tener atributos de tipo numéricos, como también cadenas de caracteres. Los Métodos son las acciones que puede realizar dicho objeto.

Cada uno de estos nace por abstracción, lo cual es una reflexión sobre qué atributos y métodos debería tener mi objeto dependiendo de mi objetivo.

Una clase es un molde o plantilla para ir creando nuevos objetos, la acción de ir creando objetos a partir de una clase se llama instanciar, lo cual a los objetos también los podemos denominar instancia. (Cervantes Ojeda & Gomez, 2016, 16).



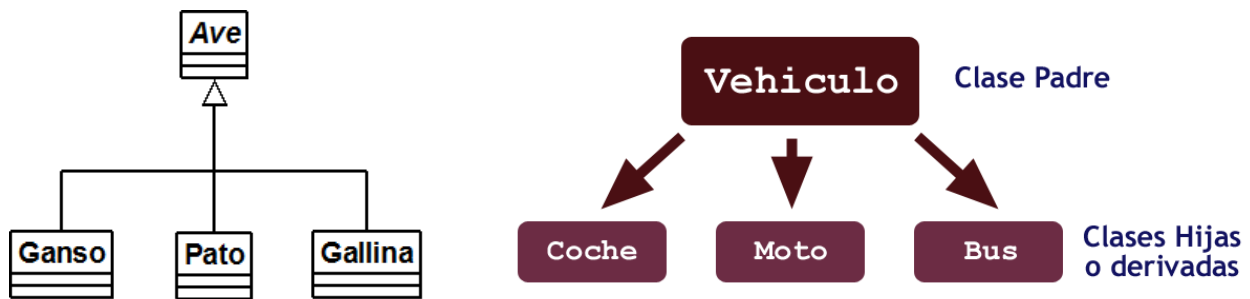
Informe Proyecto

Análisis y diseño orientado a objetos

Universidad Técnica Federico Santa María

2.-¿Qué es herencia, polimorfismo, encapsulamiento?. Explique cada una.

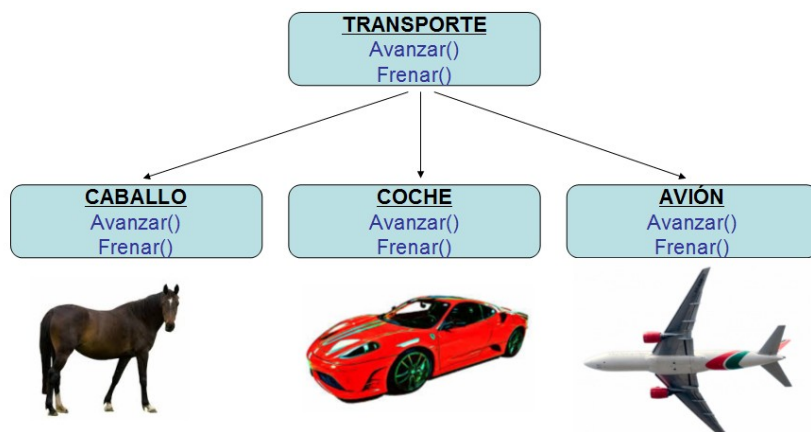
Herencia: La herencia es una relación entre clases, en las que hay una clase “padre” llamada superclase y una o más clases hijas especializadas las que se denominan subclases. (Cervantes Ojeda & Gomez, 2016, 64) La ventaja de la herencia es la capacidad para definir atributos y métodos nuevos para la subclase. lo que permite crear una estructura jerárquica de clases cada vez más especializada, lo cual facilita al momento de especializar una clase existente sin partir de cero.



Polimorfismo: El polimorfismo es una capacidad imprescindible de algunos lenguajes para que cuando se llame al mismo método desde diferentes objetos, cada uno de esos objetos pueda responder al mensaje o a la llamada de manera diferente.

Se dividen en dos niveles:

- Polimorfismo Estático: Es aquél en el que los tipos a los que se aplica el polimorfismo deben ser explicitados y declarados uno por uno antes de poder ser utilizados.
- Polimorfismo dinámico: Es un código que no contiene ninguna especificación sobre los tipos de datos en los que trabaja. Por lo tanto, se puede utilizar para todos los tipos de datos compatibles.

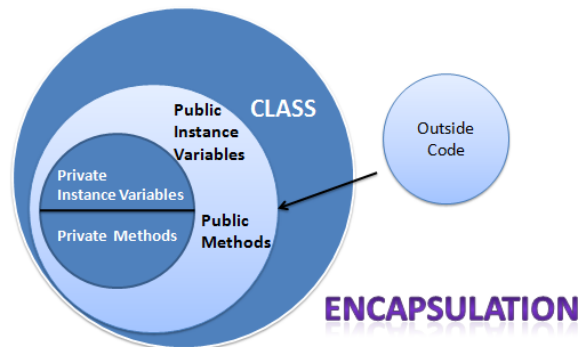


Informe Proyecto

Análisis y diseño orientado a objetos

Universidad Técnica Federico Santa María

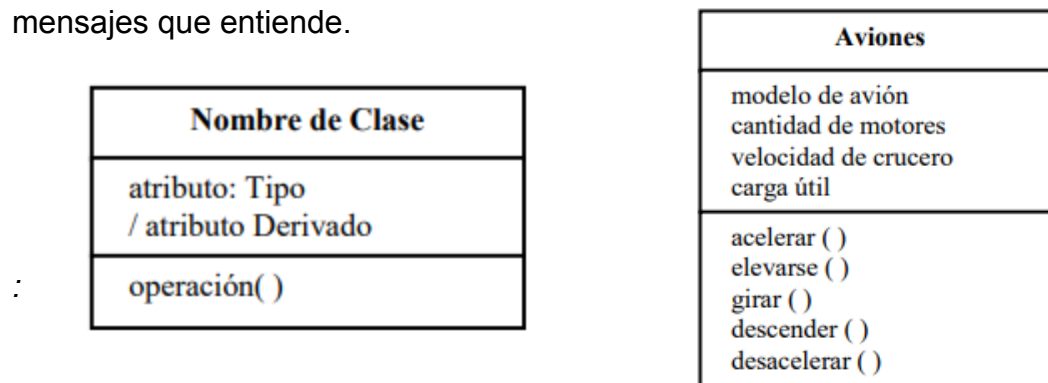
Encapsulamiento: El encapsulamiento en la programación orientada a objetos es cuando limitamos el acceso o damos un acceso restringido de una propiedad a los elementos que necesita un miembro y no a ninguno más. El elemento más común de encapsulamiento son las clases, donde encapsulamos y englobamos tanto métodos como atributos. (*Encapsulamiento En Programación Orientada a Objetos*, 2019)



3.-¿Qué son los diagramas UML?. Explique con imágenes y palabras que es el diagrama de clases el de secuencia. Y el diagrama de procesos de negocio BPMN.

El lenguaje unificado de modelado (UML) sirve para modelar, es decir diseñar antes de su implementación, lo cual nos permite especificar cuales son las características de un sistema antes de su construcción y a partir de esos modelos se pueden formar los sistemas diseñados. Este lenguaje se centra en la representación gráfica de un sistema.

Diagrama de clases: Es uno de los diagramas más comunes, es un diagrama estático, es decir, nos va a decir cuales son los componentes que se tienen, cómo se relacionan entre sí, pero no muestra información temporal. Una clase es representada por un rectángulo que posee tres divisiones: Nombre de la clase, atributos que tiene y mensajes que entiende.

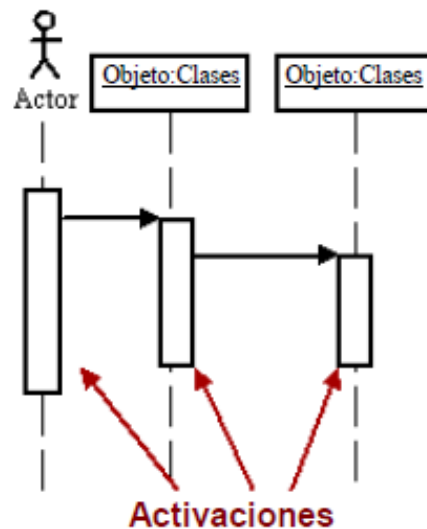


Informe Proyecto

Análisis y diseño orientado a objetos

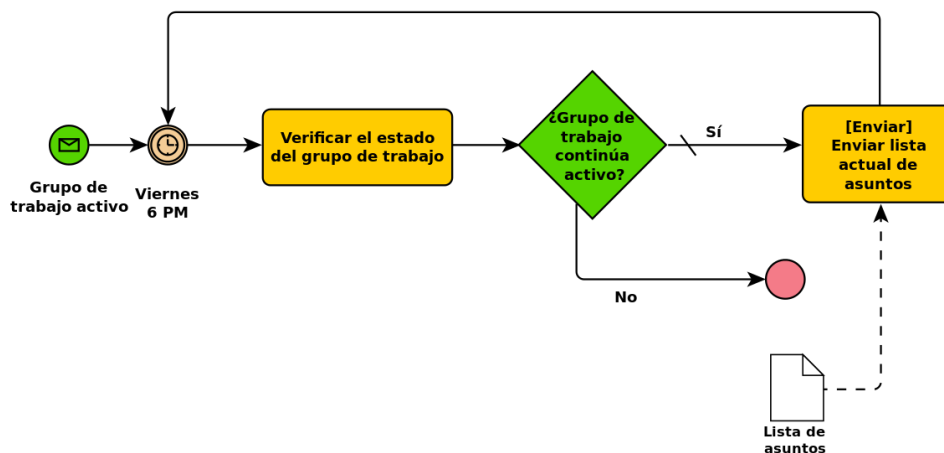
Universidad Técnica Federico Santa María

Diagramas de secuencia: Son usados para modelar la interacción entre objetos de un sistema según UML. Este muestra la forma en que un grupo de objetos interactúan entre sí a lo largo del tiempo, que toma dicha función en un sistema de software. Está constituido por objetos, mensajes y líneas de vida.



BPMN o Business Process Model and Notation: BPMN es un lenguaje visual para diagramar procesos de una manera estándar sin ambigüedades.

A partir de los diagramas UML, se inventaron la notación BPMN, que también sirve para representar procesos con la ventaja de tener muchos más símbolos disponibles que nos permiten representar situaciones típicas de los procesos, para finalmente tener un diagrama con menos símbolos, más fácil de entender y por lo tanto se podría representar procesos muchos más complejos.





4.- Indicar y explicar el problema(del cliente) a abordar.

En la Universidad Santa Maria de la sede Viña del Mar, nos hemos dado cuenta de la existencia de una problemática, en la cual se enfoca principalmente en alumnos recién ingresados, nuevos docentes y empleados. Esta problemática se ve a lo largo de todo el año pero con más frecuencia al inicio del año universitario.

Esta problemática trata de no poder identificar los edificios y sus respectivas salas, debido a que la Universidad no nos brinda mayor información sobre la ubicación de estas, más que un mapa en la entrada.

5.- Hable del target o público objetivo, aquel que tiene el problema o necesidad de software, ya sea una porción de mercado o la empresa cliente.

El público objetivo o quienes son principalmente afectados por el problema anteriormente descrito son estudiantes, con un rango de edad de entre 18 a 28 años, y que asisten a jornadas presenciales dentro de la sede Viña del Mar. Principalmente se caracterizan por desconocer las instalaciones y estar recientemente ingresados a la Universidad.

6.-¿Qué proyecto de software usted propone para ayudar a resolver/afrontar/abordar este problema?. Explique su iniciativa. ¿Cuál es el factor diferenciador de la competencia?

El proyecto de software que nosotros proponemos para afrontar y solucionar nuestra problemática dentro de la Universidad consta de la creación de una aplicación móvil, la cual será un mapa con características de GPS, con el objetivo de facilitar el reconocimiento de las salas y sus respectivos horarios con el profesor encargado con el fin de ayudar a orientarse y tener una mejor referencia sobre los edificios y la infraestructura dentro de la Universidad.

Esta aplicación puede ser más útil para alumnos, invitados, docentes y empleados recién ingresados, como también puede ser útil para personas que llevan más tiempo dentro de la Universidad .

El factor que nos diferencia es nuestro proyecto digitalizado, el cual permite una mayor fuente de información, más amigable y cercana con el usuario.



Informe Proyecto

Análisis y diseño orientado a objetos

Universidad Técnica Federico Santa María

7- ¿Qué modelo de desarrollo de software utilizará?. Explique cómo aplicará el modelo e indique las razones de su elección.

El modelo de desarrollo de software que nosotros utilizaremos será Kanban, debido a que consideramos que es el modelo que mejor se adapta a cómo desarrollaremos nuestro proyecto.

Nuestro Kanban se dividirá en cuatro columnas:

- ❖ Tareas por hacer: En esta columna se encontrarán los requerimientos, los cuales posteriormente se asignan a cada persona del equipo de trabajo.
- ❖ Tareas en proceso: Una vez asignadas las tareas por hacer, cada integrante deberá iniciar su labor.
- ❖ Complementar: En esta sección se hará una reunión con las tareas finalizadas, para luego poder hacer un perfeccionamiento de estas, apoyándose con sus otros compañeros de equipo, para así lograr una versión mejorada y más completa.
- ❖ Finalizado: En la última columna el equipo de trabajo concluye con sus labores y se procede a entregar el proyecto al cliente.

Tareas por hacer	Tareas en proceso				Complementar	Finalizado
	persona 1	persona 2	persona 3	persona 4		



Informe Proyecto

Análisis y diseño orientado a objetos

Universidad Técnica Federico Santa María

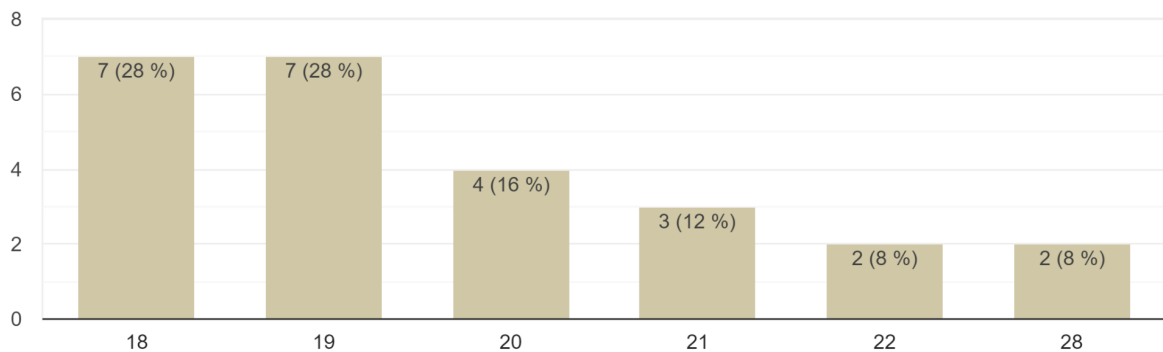
8.- Aplique al menos una técnica para levantar requerimientos y las razones de su elección. Explique muy detalladamente cómo aplicó dicha técnica.

Para levantar requerimientos se utilizará una breve encuesta con 10 preguntas para poder obtener respuestas generales y poder tener los requerimientos funcionales y no funcionales que podrían aportar valor a la propuesta de software.

Aplicamos la encuesta a través de un formulario de Google en el cual contestaron 26 personas (26 estudiantes). A continuación se presentan datos de los encuestados, las preguntas y respuestas resumidas.

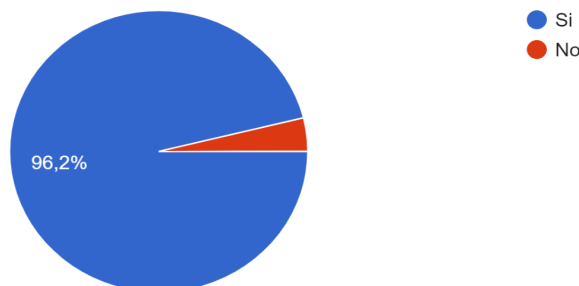
Edades de los encuestados.

- Eje vertical: Cantidad de personas.
- Eje horizontal: Edades.



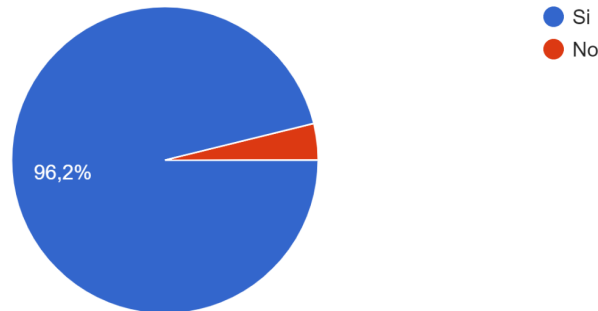
Preguntas y respuestas.

A) ¿Has tenido problemas para encontrar algún edificio/departamento dentro de la Universidad?





B) ¿Crees que los demás estudiantes, profesores y empleados comparten la misma problemática?



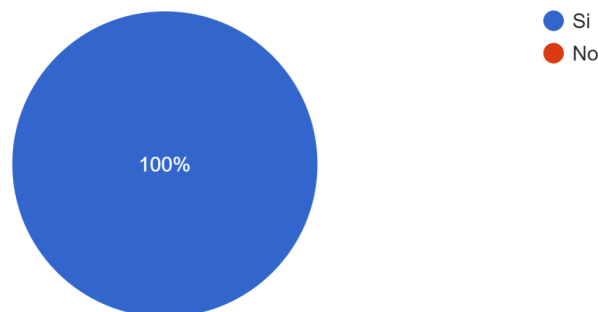
C) ¿Cuál fue la primera referencia que viste dentro de la Universidad para saber dónde estaba cada edificio? (página, foto, etc) ¿te sirvió esta referencia?

La mayor parte de los encuestados utilizaron el mapa de la entrada, que aparentemente no fue muy efectivo, en menor cantidad hubieron quienes utilizaron fotos para localizar zonas dentro de la universidad.

D) ¿Cómo mejorarías estas referencias físicas para que puedan ser de mayor utilidad?

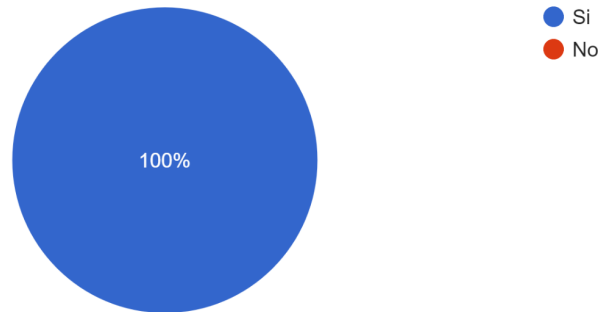
La mayor parte de los encuestados prefieren reparación y/o mantenimiento en el mapa de la entrada y mejoras en las referencias que se encuentren físicamente dentro de la universidad, añadiendo señalizaciones e indicaciones.

E) ¿Posees algún dispositivo portátil (celular o tablet) con conexión permanente a internet?

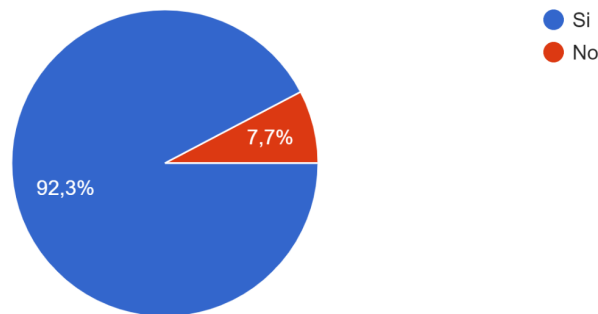




F) Si se crea una Aplicación móvil la cual te ayude a orientarte mejor dentro de la universidad y su infraestructura. ¿La usarías?



G) ¿Crees que esta aplicación servirá para terminar con la problemática?



H) ¿Qué funciones y/o características esperarías que incluya esta aplicación?

Todos coinciden en incluir un mapa detallado, con referencias gráficas y información correcta, complementado también cada edificio con su respectiva información relevante (profesores, salas, facultades y horarios).



Informe Proyecto

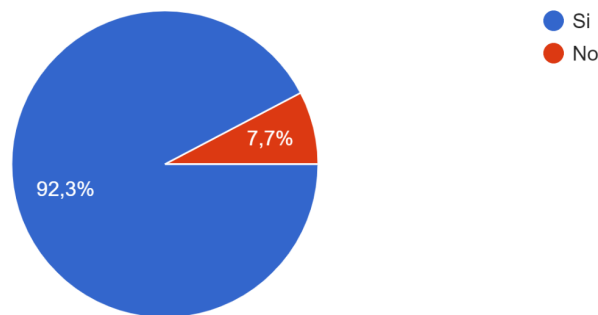
Análisis y *diseño orientado a objetos*

Universidad Técnica Federico Santa María

I) Con respecto al formato del mapa ¿Prefieres un mapa detallado y con referencias visuales o uno con solo la información más relevante y simplificada? Como el nombre y letra de cada edificio. ¿Cuál es la razón de tu preferencia?

Lo más recalcado aquí fue el tener un mapa detallado y con referencias visuales, siendo también nombrado el detallar información sobre cada edificio.

J) ¿Consideras útil que en la aplicación puedas acceder a otro tipo de información como por ejemplo horarios de clases y eventos dentro de la universidad?





Informe Proyecto

Análisis y *diseño orientado a objetos*

Universidad Técnica Federico Santa María

9.- Aplique alguna técnica para priorizar requerimientos: Puede ser RICE, Moscow u otra a elección.

Nuestra técnica para priorizar requerimientos será Moscow. Nuestros requerimientos Funcionales y no funcionales son:

1. Mapa en forma de GPS que esté bien detallado, donde nos indique los pasos a seguir para llegar a nuestro destino.
2. Mostrar horarios de las salas y profesores encargados.
3. Horarios de las clases que se están impartiendo en tiempo real.
4. Imágenes del interior de los diversos edificios.
5. Función de búsqueda para encontrar los edificios.
6. Función de poder crear los horarios de clases asignando con su respectiva sala.
7. Sección para filtrar según carrera.
8. Recordatorio tipo calendario de las clases del día.
9. Departamento de cada carrera.
10. Información importante en relación a la universidad y sus actividades.
11. Muro de consultas, con preguntas más comunes.
12. Interfaz simple y didáctica para el usuario.

Must Have (Debe tener):

- Mapa en forma de GPS que esté bien detallado, donde nos indique los pasos a seguir para llegar a nuestro destino.
- Mostrar horarios de las salas y profesores encargados.
- Función de búsqueda para encontrar los edificios.
- Departamento de cada carrera.
- Interfaz simple y didáctica para el usuario.

Should Have (Debería tener):

- Sección para filtrar según carrera.
- Imágenes del interior de los diversos edificios.

Could have (Podría tener):

- Muro de consultas con preguntas más comunes.
- Información importante en relación a la universidad y sus actividades.



Informe Proyecto

Análisis y diseño orientado a objetos

Universidad Técnica Federico Santa María

Won't have but would like to in the future (No tendrá pero me gustaría en el futuro):

- Horarios de las clases que se están impartiendo en tiempo real.
- Función de poder crear los horarios de clases asignando con su respectiva sala.
- Recordatorio tipo calendario de las clases del día.

10.- Indique los requerimientos funcionales y no funcionales.

Requerimientos funcionales:

Referencia	Comentario
RF-01	La aplicación deberá cumplir la función de un mapa en forma de GPS, para llevar a la persona hacia su destino y/o brindarle información.
RF-02	La aplicación deberá otorgar referencias visuales (fotos y posición en el mapa) para una mayor comprensión de esta.
RF-03	La aplicación deberá permitir ver los horarios de cada ramo, la sala asignada y el profesor encargado.
RF-04	La aplicación deberá mostrar los profesores encargados de cada asignatura.
RF-05	La aplicación deberá poder aplicar filtros según facultad seleccionada.
RF-06	La aplicación deberá permitir una búsqueda por edificio en específico.
RF-07	La aplicación deberá permitir acceso a un muro de consultas, con sus respectivas respuestas.
RF-08	La aplicación deberá contener información importante respecto al edificio seleccionado.



Informe Proyecto

Análisis y *diseño orientado a objetos*

Universidad Técnica Federico Santa María

Requerimientos No funcionales:

Referencia	Comentario
RF-01	La interfaz para el usuario debe ser lo más simple e intuitiva posible, permitiendo abarcar un mayor rango de personas.



References

Cervantes Ojeda, J., & Gomez, M. D. C. (2016). *Introducción a la programación orientada a objetos* (Vol. 1). Ricardo López Gómez.

Encapsulamiento en programación orientada a objetos. (2019, September 14).

Netmentor. Retrieved September 10, 2022, from

<https://www.netmentor.es/entrada/encapsulamiento-poo>

Morero, F. (2000). *Introducción a la OOP* (Versión 1.0.0 ed.).