Examen 2

Contenido

[Impacto de riesgos 2](#_Toc157524044)

[Aplicación de la administración de riesgos 3](#_Toc157524045)

[Diagrama administración de riesgos 3](#_Toc157524046)

[Análisis de riesgos prioritarios 4](#_Toc157524047)

[¿Cuáles son los riesgos más comunes que has experimentado en tu proyecto? ¿Cómo los has resuelto? 5](#_Toc157524048)

[Prueba unitaria 5](#_Toc157524049)

[Casos de prueba 6](#_Toc157524050)

[Patrón de diseño 7](#_Toc157524051)

[¿Qué te ha permitido realizar una planificación de tu proyecto? ¿Cómo sería tu proyecto sin planificación? 8](#_Toc157524052)

[Componentes y actores 8](#_Toc157524053)

[Costo del proyecto 9](#_Toc157524054)

[Aceptación del cliente 12](#_Toc157524055)

[Mejoras del curso 12](#_Toc157524056)

# Impacto de riesgos

En esta sección se muestra el análisis del impacto financiero del riesgo de la siguiente tabla, es decir, cuando costaría este riesgo si se convierte en un problema.

Tabla 1. Descripción del riesgo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Riesgo** | **Probabilidad** | **Impacto** | **Estrategia de mitigación** | **Estrategia de contingencia** |
| Contraer alguna enfermedad respiratoria | Alta | Alto | Uso de cubrebocas en espacios públicos | Tener listo un tratamiento para sanar lo más pronto posible |

En el momento en el que este riesgo se convierte en un problema tendrá un gran impacto financiero el cual se puede dividir en los siguientes ámbitos:

1. **Impacto económico personal.** Los medicamentos para controlar las enfermedades respiratorias son caros. Debido a esto el contagiarme con alguna enfermedad respiratoria representa un gasto fuerte de dinero para poder solucionar el problema.
2. **Impacto económico del proyecto.** Una consecuencia de que este riesgo se convierta en un problema es que los síntomas evitan que pueda trabajar por al menos un par de días, en caso de que sea grave este tiempo se puede convertir en un par de semanas. Esta pérdida de tiempo se convierte en un incumplimiento en las entregas prometidas lo que representa que el cliente puede dejar de pagar los montos establecidos.
3. **Impacto académico.** Faltar a la escuela por alguna enfermedad no representa un impacto financiero directo. Sin embargo, representa un impacto en las tareas que se hacen de todas las materias por lo que a largo plazo también se convertirá en un impacto financiero.

# Aplicación de la administración de riesgos

La administración de riesgos la he aplicado de una forma sencilla, pero hasta el momento muy eficaz. En el curso pudimos ver las etapas de la administración de riesgos; estos pasos fueron los que apliqué dentro del proyecto. Los pasos son sencillos de seguir, pero requieren de una gran disciplina y proactividad para evitar trabajar en problemas que no sabemos de dónde salen. A pesar de que se aplicaron todos los pasos de la administración de riesgos en el proyecto los pasos que en lo personal representaron un cambio más significativo fueron: identificar, analizar y priorizar los riesgos. La importancia de identificar los riesgos radica en diferenciar entre problema y riesgo; lo que se busca con la administración de riesgos es evitar que los riegos que ya se identificaron no se vuelvan problemas durante el desarrollo. Analizar los riesgos resultó ser una herramienta extremadamente poderosa ya que me permitió cuantificar la probabilidad de que un riesgo ocurra y en ese caso el impacto que puede ocasionar. Por último, está la priorización, con base en la cuantización del paso anterior nos ayuda a tener un plan de acción básico de acuerdo con que riesgo tiene una mayor probabilidad de que ocurra o el que tenga más impacto.

# Diagrama administración de riesgos

En siguiente figura se muestra un diagrama Scrum genérico. Él cual servirá como una base para indicar como se aplicó la administración de riesgos en el proyecto.

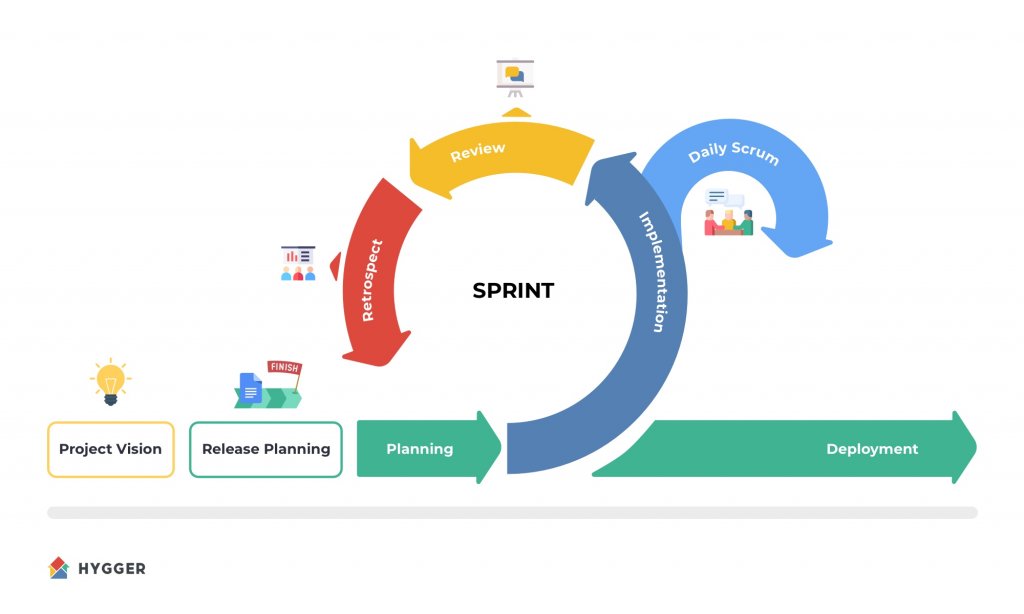


Figura 1. Diagrama Scrum

Como se puede ver en la figura la primera etapa es la visión del proyecto. De esta visión es en donde se crean las historias de usuario; idealmente en esta etapa se deben tener una identificación inicial de riesgos la cual consiste en: identificación, análisis y priorización. Sin embargo, la administración de riesgos se vio después de tener identificadas las historias de usuario por lo que esta identificación inicial de riesgos se tuvo de manera retroactiva. De tal modo que se usó la experiencia obtenida del proyecto para poder generar un análisis y priorización más detallado de los riesgos.

En la etapa de planeación ha sido en la que mejor se ha puesto en práctica la administración de riesgo. Debido a que con el cliente se ha tomado la decisión de llevar a cabo Sprints la administración de riesgos me ha permitido revisar los riesgos asociados a las historias de usuario de cada Sprint. Y así poder tomar medidas para evitar que estos riegos se vuelvan problemas durante el Sprint.

Por último, queda la etapa de liberación la cual debido a los tiempos manejado aún no ha comenzado hasta el día en el que se entrega este documento. La administración de riesgos me permite revisar el estado de los riesgos y buscar nuevos riesgos que se pueden ver implicados en la liberación del sistema.

# Análisis de riesgos prioritarios

En la siguiente tabla se pueden ver los riegos que tienen más impacto en el proyecto

Tabla 2. Riesgos identificados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Riesgo** | **Probabilidad** | **Impacto** | **Estrategia de mitigación** | **Estrategia de contingencia** |
| Contraer alguna enfermedad respiratoria | Alta | Alto | Uso de cubrebocas en espacios públicos | Tener listo un tratamiento para sanar lo más pronto posible |
| No poder dedicarle el tiempo necesario al proyecto por ser final de trimestre | Media | Alto | Ordenar las tareas y proyectos de otras materias de acuerdo con su prioridad | Tener compañeros de equipo en las otras materias como apoyo para las tareas |
| La computadora en la que desarrollo se descomponga | Baja | Alto | No descargo cosas desconocidas de internet durante proyectos importantes. Además de realizar chequeos preventivos | Todo el código fuente y documentación se encuentra respaldada en un repositorio en la nube |
| Perder la computadora por un robo o asalto | Alta | Alto | No sacar la computadora en lugares públicos o innecesariamente | Todo el código fuente y documentación se encuentra respaldada en un repositorio en la nube |
| Que el proyecto no se pueda implementar en producción | Baja | Alto | Tener las versiones más compatibles de todos los elementos que se usan en el proyecto | Tener en un repositorio en la nube con la última versión funcional del proyecto |

Los riegos mostrados en la tabla anterior tienen un impacto alto debido a que si se convierten en un problema ponen en riegos la entrega de avances significativos en las fechas establecidas. Por lo tanto, es de vital importancia tener estrategias de contingencia en caso de que uno o más riesgos se vuelvan problemas durante el desarrollo del proyecto

# ¿Cuáles son los riesgos más comunes que has experimentado en tu proyecto? ¿Cómo los has resuelto?

El riesgo más común es contraer alguna enfermedad respiratoria. Este riesgo es del que más me tengo que cuidar ya que hay dos factores que hacen que la posibilidad que se convierta en un riesgo: la primera es que el clima de esta época del año ha sido demasiado frío; este clima es particularmente dañino para mi por padecer de asma. El segundo factor es tener más contacto con personas enfermas. Estos dos factores provocan que cada día tenga llevar a cabo medidas para evitar algún contagio.

Afortunadamente lo he podido resolver usando cubrebocas en lugares públicos, así como usar ropa lo suficientemente abrigadora como para que el frío no me afecto. En las ocasiones en las que he sentido que me empiezo a enfermar acudo a un tratamiento especial que tomado a tiempo evita que la enfermedad empeore más de lo necesario. Estas son las medidas que me han permitido evitar que el riesgo se vuelva problema y en los casos en los que no se puede evitar se ha logrado mitigar el impacto del problema lo más posible.

# Prueba unitaria

Para realizar una prueba unitaria en un proyecto web creado con Django, seguí los siguientes pasos:

1. **Configurar el entorno de prueba.** Consiste en tener configurado un entorno de prueba para la aplicación Django. Django proporciona un conjunto de herramientas para facilitar las pruebas unitarias, incluyendo un framework de pruebas integrado.
2. **Estructura de pruebas.** Django sigue una estructura de pruebas dentro de cada aplicación. En el directorio de tu aplicación, generalmente llamado *tests.py*, es donde escribirás tus pruebas unitarias.
3. **Importar las herramientas de prueba.** En el archivo *tests.py*, comienzo importando las herramientas necesarias de Django para escribir las pruebas. Algunas de las herramientas comunes que se necesitan son *TestCase* y las funciones de *assert* para comprobar los resultados esperados.
4. **Escribir las pruebas.** Se definen clases de prueba que hereden de *django.test.TestCase*. Dentro de estas clases, se definen métodos de prueba que verifican el comportamiento deseado del código.
5. **Ejecutar las pruebas.** Se utiliza el administrador de comandos de Django para ejecutar las pruebas. Se pueden ejecutar todas las pruebas del proyecto o pruebas específicas de una aplicación o módulo.
6. **Interpretación de los resultados.** Después de ejecutar las pruebas, Django proporciona un resumen de los resultados.

# Casos de prueba

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Nombre** | **Precondiciones** | **Valores de entrada** | **Valores aceptados de salida** |
| 1 | Crear usuario | Contar con un correo electrónico, ingresar al sitio | Nombre, primer apellido, segundo apellido, fecha de nacimiento, correo electrónico, contraseña | Mensaje de confirmación, usuario creado correctamente |
| 2 | Iniciar sesión | El actor tiene un nombre correo y contraseña válidas | Correo electrónico, contraseña | El actor inicia sesión en el sistema y accede al perfil correspondiente |
| 3 | Cerrar sesión | El actor tiene una sesión activa | Ninguno | El actor cierra sesión y se redirige a la página de inicio |
| 4 | Consultar perfil | El actor tiene sesión activa en el sistema | Ninguno | El sistema muestra la información del perfil del actor |
| 5 | Modificar perfil | El actor tiene una sesión activa en el sistema  El actor proporciona los datos a modificar | Nombre, primer apellido, segundo apellido, contraseña | Los datos del perfil del actor se actualizar en el sistema |

# Patrón de diseño

El patrón de diseño MVT (Modelo, Vista, Template) es una variante del patrón de diseño MVC (Modelo-Vista-Controlador) que se utiliza en el desarrollo de aplicaciones web, especialmente en el contexto de frameworks como Django, que es un marco web de alto nivel para el desarrollo rápido de aplicaciones en Python.

A continuación, se describe cada componente del patrón MVT:

**Modelo (Model):**

El Modelo representa la capa de datos y la lógica de negocios. Se encarga de gestionar y manipular los datos, así como de realizar operaciones relacionadas con la lógica de la aplicación. En Django, los modelos se definen mediante clases Python que representan las tablas de la base de datos y definen la estructura y comportamiento de los datos.

**Vista (View):**

La Vista es responsable de presentar los datos al usuario y de manejar las interacciones del usuario.

En Django, las vistas son funciones o clases que toman una solicitud HTTP y devuelven una respuesta HTTP. Las vistas pueden renderizar plantillas (***templates***) para presentar datos al usuario.

**Plantilla (Template):**

La Plantilla se encarga de la presentación visual y define la estructura y el formato de la interfaz de usuario. En Django, las plantillas son archivos que contienen código HTML con la capacidad de incrustar expresiones y bloques de control de flujo utilizando la sintaxis de las plantillas de Django. Las plantillas se utilizan para generar el contenido dinámico que se envía al usuario.

El flujo de trabajo típico en un patrón MVT en Django sería el siguiente:

1. El usuario realiza una solicitud HTTP, como visitar una página en un navegador.
2. La solicitud es manejada por una vista en Django, que puede realizar operaciones en el modelo y luego renderizar una plantilla.
3. La plantilla se llena con datos provenientes del modelo y se presenta al usuario como una respuesta HTTP.
4. El usuario ve la interfaz de usuario generada y puede interactuar con ella.
5. Si el usuario realiza acciones que generan nuevas solicitudes, el ciclo se repite.

En resumen, el patrón MVT en Django comparte muchos conceptos con el patrón MVC, pero la nomenclatura es adaptada para reflejar las peculiaridades de los marcos de desarrollo web y la forma en que Django organiza y maneja las capas de una aplicación.

# ¿Qué te ha permitido realizar una planificación de tu proyecto? ¿Cómo sería tu proyecto sin planificación?

El elemento clave que me ha permitido realizar una planificación de proyecto fue la etapa de análisis y diseño. Esta etapa me permitió definir de una manera eficaz todas las necesidades del cliente; tener la flexibilidad de poder refinar las historias de usuario de la mano del cliente facilitó muchísimo la planificación. Las historias de usuario fueron los cimientos de mi planificación ya que me permitieron visualizar el diseño del sistema y con esto la mejor forma de llevar a cabo el desarrollo, las pruebas y la implementación. De no ser por la planificación el proyecto no tendría pies ni cabeza, sería una pesadilla organizar los módulos que comprenden el sistema. Sin planificación sería imposible darle al cliente avances significativos dentro de fechas de entrega reales.

# Componentes y actores

Los actores son la representación de los usuarios al momento de interactuar con el sistema. En el proyecto hay tres actores:

* **Administrador.** Es la persona encargada de gestionar a los usuarios del sistema. Tiene la capacidad de asignar roles, gestionar la base de datos y administrar aspectos generales del sistema.
* **Asesor.** Es el profesional especializado en nutrición y/o actividad física que brinda asesoramiento personalizado a sus clientes. Puede crear planes alimenticios y rutinas, realizar seguimientos del progreso de sus clientes y enviar invitaciones a usuarios del sistema para poder asesorarlos.
* **Cliente.** Usuario que recibe asesorías de un asesor. Puede acceder a los planes alimenticios y rutinas que se le asignen, registrar su actividad física y su ingesta calórica, así como consultar los reportes de progreso generados por su asesor.

Los principales componentes del sistema serían los siguientes:

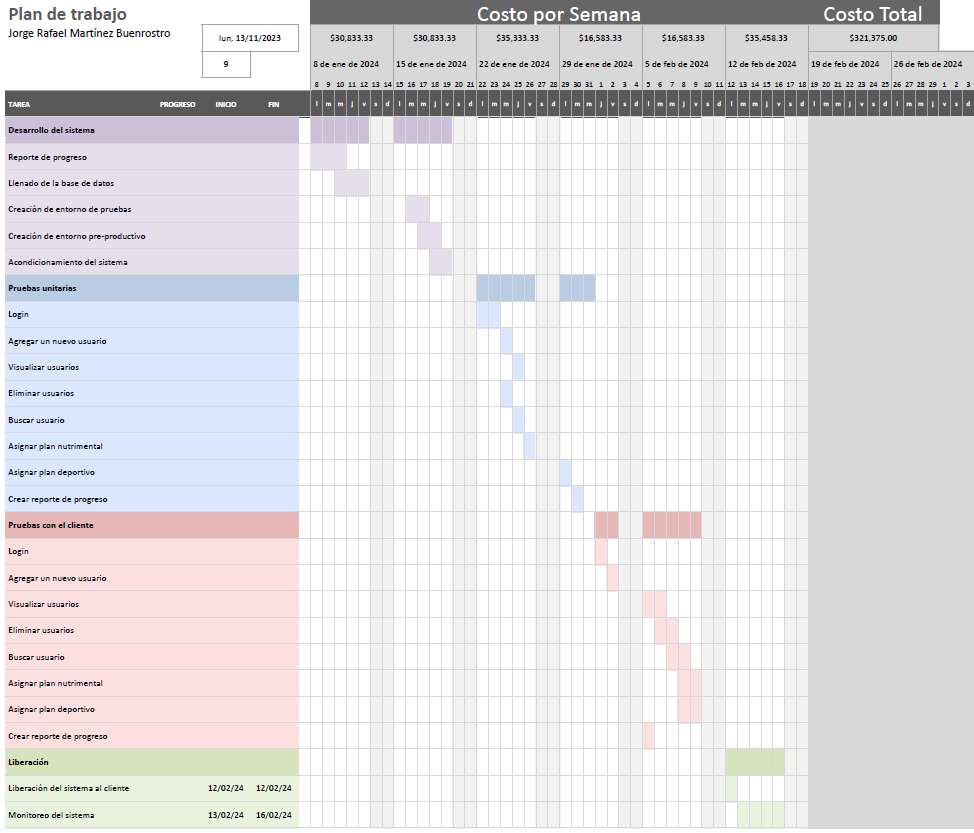
1. **Framework Django. D**jango es un framework web de alto nivel que facilita el desarrollo rápido y limpio de aplicaciones web en Python. Proporciona una estructura sólida para el desarrollo, incluyendo manejo de URL, vistas, plantillas, formularios y administración de bases de datos.
2. **Base de datos.** La base de datos del sistema se construyó usando PostgreSQL. El cual es un sistema de gestión de bases de datos relacional de código abierto y potente. Se integra bien con Django.
3. **Modelos Django.** Los modelos en Django son clases de Python que representan las tablas en la base de datos. Cada modelo define los campos de la tabla y sus tipos de datos, así como también define relaciones entre tablas si es necesario.
4. **Vistas Django.** Las vistas son funciones de Python que procesan las solicitudes web y devuelven las respuestas. En un proyecto Django, las vistas son responsables de interactuar con los modelos, realizar lógica de negocio y devolver datos a las plantillas.
5. **Plantillas Django.** Las plantillas son archivos HTML que contienen la presentación de su aplicación web. Django utiliza su sistema de plantillas para generar HTML dinámico, permitiendo la inserción de datos desde las vistas y la lógica de control.
6. **Archivos estáticos.** Estos son archivos como CSS, JavaScript e imágenes que se utilizan para la presentación y la interactividad en el lado del cliente. Django proporciona soporte para manejar archivos estáticos y servirlos de manera eficiente.
7. **Middleware Django.**  El middleware en Django es una capa de procesamiento de solicitudes y respuestas. Puede utilizar middleware para realizar tareas como autenticación, compresión de contenido, manejo de excepciones, etc.
8. **Entorno virtual.** Es una herramienta que se utiliza para crear entornos Python aislados. Es una buena práctica utilizar entornos virtuales para cada proyecto Django para gestionar las dependencias de manera eficiente.
9. **Rutas URL.** En el archivo `urls.py`, se definen las URL de la aplicación y se mapean a las vistas correspondientes que manejan esas URL.

# Costo del proyecto

En las siguientes figuras se muestra el diagrama de Gantt del proyecto. En el cual se puede observar el costo por semana de cada etapa del desarrollo, así como el costo final. Estos costos se actualizaron con base en los cambios realizados durante el desarrollo

Tabla

Descripción generada automáticamente con confianza media



# Aceptación del cliente

Si tuviéramos que realizar una presentación al cliente/usuario en una semana, ¿consideras que tu proyecto sería aceptado por el cliente?

La respuesta corta es que no sería aceptado. Ahora bien, esto se debe a varios motivos que tienen que ver con la naturaleza del proyecto. Este proyecto está enfocado para un cliente real que encontré al inicio del trimestre. Toda la parte de la toma de requerimientos se llevó conforme se veía en clase; sin embargo, ya que este proyecto lo estoy desarrollando completamente se crearon dos planes de trabajo a partir de la fase de análisis. El plan de trabajo presentado en la clase contempla un desarrollo de cuatro semanas, este tiempo fue arbitrariamente elegido para poder cumplir con las fechas del trimestre. Mientras que el plan de trabajo del cliente es más extenso principalmente por la metodología usada. Para poder asegurar al cliente como un equipo de desarrollo tan pequeño se acordó tener Sprints de dos semanas de los cuales se entregan pequeñas mejoras respecto al Sprint anterior. Además, se agrega un poco más de tiempo para tener mejores pruebas lo que me permite entregar un trabajo de mayor calidad y facilitar la implementación productiva del sistema.

Por estos motivos es que si en una semana el cliente quisiera una presentación del sistema con lo que se tiene para la clase es muy probable que no lo acepte.

# Mejoras del curso

Con base en mi experiencia durante este curso puedo proponer algunas mejoras para futuros trimestres. La primera sería tener un horario fijo y estable de clases. Se que este trimestre hubo algunas situaciones extraordinarias que afectaron los horarios de clase y aunque se buscaron soluciones siento que esta inestabilidad afectaba el hilo que se lleva entre cada clase.

Otra mejora es que se limpie la carpeta de recursos; tiende a ser algo confuso tener múltiples copias del mismo archivo o tener en la misma carpeta diferentes formatos del mismo material. Tener que buscar entre estos archivos hacía que más lento el proceso para hacer alguna actividad.

Por último, a pesar de que muchos de los temas son tocados en las distintas etapas vistas en clase. Algunas de las presentaciones tendían a ser muy repetitivas. Esto hacía que algunas clases fueran algo repetitivas y se sintieran como algo que se podría evitar