

Trabajo Final Algoritmia y Programación 2025-1

Profesor: Victor Hugo Mercado Ramos

Curso: Algoritmia y Programación

Contenido

Descripción del problema a solucionar – El Hotel BuenDescanso.....	2
El Hotel BuenDescanso.....	2
Objetivo:	3
Reglas:	3
Actas de entendimiento y compromiso.	3
Objetivo de entendimiento y compromiso	3
Alcance.....	3
Definiciones (Entregables)	3
Procedimientos (Entregables)	3
Seguimiento y Evaluación (Entregables).....	4
Resolución de Conflictos	4
Actividades	4
Registrar Huésped.....	5
Realizar Reserva de Habitación.....	6
Registrar Salida (Check-Out).....	6
Administrador:	7
El módulo de administración deberá ofrecer los siguientes reportes:.....	7
Visualización de Reportes Gráficos con Matplotlib	7
Conversación inicial con el profesor: ¡Haciendo las preguntas correctas!.....	8
Objetivos y Propósito	8
Funcionalidades y Requisitos	9
Prioridades y Alcance	9
Limitaciones y Restricciones	9
Interfaz y Experiencia del Usuario	9
Pruebas y Calidad	9
Riesgos y Mitigación	9
Colaboración y Comunicación.....	9
Registro de Actividades (Log de Eventos)	10
Entregables	10

1. Integrantes:	10
2. Vínculos académicos y descripción:	10
3. Nombre del proyecto y detalles:	10
4. Licencia del software:	11
5. Reporte de visión:	11
6. Especificación de requisitos:	11
7. Plan de proyecto:	11
8. Plan de versionado:	11
9. Algoritmo:	11
10. Manual de usuario:	11
11. GitHub:	12
12. Sustentaciones:	12
13. Fechas y Documentos:	12

Descripción del problema a solucionar – El Hotel BuenDescanso

El Hotel BuenDescanso es un establecimiento que presta sus servicios de alojamiento a visitantes y turistas del sector céntrico de la ciudad. Este hotel ofrece hospedaje exclusivamente en habitaciones individuales y suites; no se permite el alojamiento de mascotas. El Hotel BuenDescanso requiere de sus habilidades como un equipo de programación de la respetada y bien ponderada Universidad de Antioquia para crear un software de consola en Python que permita gestionar huéspedes, realizar reservas, registrar ingresos y salidas, generar cobros, facturas, reportes administrativos, entre otras funcionalidades.

El Hotel BuenDescanso cuenta con tres recepcionistas que trabajan turnos de 8 horas para atender a los huéspedes durante todo el día, cubriendo así una jornada continua de 24 horas. El programa debe registrar huéspedes, realizar reservas de habitaciones, registrar ingresos y salidas de los huéspedes, y generar los respectivos reportes administrativos.

Para solucionar este problema, los estudiantes en grupos de tres integrantes deberán crear un programa que se encargue de gestionar el Hotel BuenDescanso, mediante entregas programadas y detalladas que permitan alcanzar el propósito de desarrollo. El programa debe ser entregado en un repositorio de GitHub, donde el líder del equipo será el encargado de crear y gestionar el espacio de desarrollo.

El Hotel BuenDescanso

El Hotel BuenDescanso cuenta con una disponibilidad de 40 habitaciones, de las

cuales 30 son estándar y 10 son suites, las cuales son de libre elección por parte de los huéspedes según disponibilidad. Actualmente, el registro de reservas y estancias se realiza de manera manual en libros físicos, donde se anotan a mano los datos del huésped, la fecha de ingreso y la duración de la estancia. Al momento del check-out, el recepcionista calcula manualmente el valor a pagar según el tipo de habitación y número de noches. El hotel, hasta el momento, no genera facturas oficiales ni documentos que certifiquen de manera formal el pago realizado por el huésped.

Objetivo:

Crear un programa de consola visualmente amigable al usuario, que permita gestionar la atención y los servicios ofrecidos por el Hotel BuenDescanso, incluyendo el registro de huéspedes, la realización de reservas, el control de ingresos y salidas, la facturación y la generación de reportes administrativos. Además, el sistema deberá permitir exportar los resultados a un archivo plano (CSV) utilizando Python para facilitar el manejo y análisis de la información

Reglas:

Actas de entendimiento y compromiso.

Los integrantes del grupo deben definir su participación y responsabilidad mediante actas de entendimiento. Estas se describen a continuación:

Objetivo de entendimiento y compromiso

Este procedimiento tiene como objetivo establecer las normas y directrices para la creación de actas de entendimiento, colaboración y responsabilidad en trabajos grupales académicos. Busca promover un ambiente de trabajo colaborativo y respetuoso, asegurando que todos los miembros del grupo comprendan y acuerden sus roles y responsabilidades.

Alcance

Este procedimiento aplica a todos los estudiantes vinculados a un equipo para la entrega del trabajo final.

Definiciones (Entregables)

- Acta de Entendimiento: Documento que detalla los objetivos comunes del grupo y las expectativas de cada miembro respecto al proyecto.
- Acta de Colaboración: Documento que especifica las metodologías de trabajo en equipo, incluyendo estrategias de comunicación y resolución de conflictos.
- Acta de Responsabilidad: Documento que asigna tareas específicas a cada miembro del grupo, estableciendo plazos y criterios de evaluación.

Procedimientos (Entregables)

- Convocatoria de Reunión: El grupo deberá convocar a una reunión inicial para discutir los objetivos del proyecto y elaborar el Acta de Entendimiento.
- Elaboración del Acta de Entendimiento: Durante la reunión, los miembros del

grupo discutirán y acordarán los objetivos y expectativas del proyecto. Esta acta será redactada por un miembro designado y luego revisada y firmada por todos los integrantes.

- **Elaboración del Acta de Colaboración:** Posteriormente, se redactará el Acta de Colaboración, donde se definirán las normas de trabajo en equipo, incluyendo los canales y frecuencia de comunicación.
- **Asignación de Responsabilidades:** Finalmente, se creará el Acta de Responsabilidad, detallando las tareas específicas de cada miembro, así como los plazos de entrega. Cada miembro del grupo deberá firmar este documento, aceptando sus responsabilidades.

Seguimiento y Evaluación (Entregables)

- **Reuniones de Seguimiento:** El grupo deberá realizar reuniones periódicas para evaluar el progreso del proyecto y hacer ajustes necesarios en las actas.
- **Revisión de Actas:** Las actas pueden ser revisadas y modificadas con el consentimiento de todos los miembros del grupo, según sea necesario para reflejar cambios en el proyecto o en la dinámica del equipo.

Resolución de Conflictos

En caso de desacuerdos o conflictos, el grupo deberá referirse al Acta de Colaboración para resolver la situación mediante los mecanismos previamente acordados. Si no se llega a una resolución, se consultará con el docente del curso para mediación.

Actividades

El programa de gestión del Hotel BuenDescanso debe realizar las actividades que se muestran en la Ilustración 1 y que se detallan a continuación:

```

      _||_
      |  |
      |__|
      |  |
      |HOTEL|
      |  |
      |__|
      |  |
      |  |
      |  |
      |  |
      |__|

=====
Bienvenido al Hotel BuenDescanso
=====
1. Registrar Huésped
2. Realizar Reserva
3. Registrar Ingreso (Check-In)
4. Registrar Salida (Check-Out)
5. Administración (Acceso restringido)
6. Salir
=====
Seleccione una opción:

```

Ilustración 1 Menú de Inicio Hotel BuenDescanso.

Registrar Huésped

El huésped que llegue al hotel será registrado en el sistema con los siguientes datos:

- Nombre, el cual debe cumplir las siguientes validaciones:
 - Longitud: No puede tener menos de tres letras.
 - Contenido: No puede contener números ni caracteres especiales.
- Apellido, el cual debe cumplir las siguientes validaciones:
 - Longitud: No puede tener menos de tres letras.
 - Contenido: No puede contener números ni caracteres especiales.
- Documento de Identidad, que debe cumplir las siguientes validaciones:
 - Longitud: Debe tener entre 3 y 15 dígitos.
 - Contenido: Solo se permiten números; no se aceptan letras ni otros caracteres.
- Correo Electrónico, el cual debe cumplir las siguientes validaciones:
 - Formato: Debe ser un correo válido, es decir, contener un '@' y un

dominio.

- Número de Teléfono de Contacto, el cual debe cumplir las siguientes validaciones:
 - Longitud: Debe ser mínimo de 7 y máximo de 15 dígitos.
 - Contenido: Solo se permiten números.

Nota:

En caso de encontrar errores en la validación de estos datos, el sistema debe informar al usuario de manera clara cuáles fueron los errores detectados. En caso de que existan múltiples errores, se podrán mostrar secuencialmente o listarlos todos de manera simultánea. Se deja a libre adopción por parte del equipo de desarrollo la forma de mostrar los errores.

Realizar Reserva de Habitación

El programa solo permitirá realizar reservas para huéspedes previamente registrados.

El equipo de desarrollo deberá consultar y evaluar el procedimiento adecuado para mostrar las habitaciones disponibles y para calcular el período de estancia de un huésped en el hotel.

El software debe:

- Validar que el huésped esté correctamente registrado.
- Validar los datos de la reserva ingresados:
 - Tipo de habitación (Estándar o Suite).
 - Fecha de ingreso.
 - Número de noches de estancia.
- Confirmar que exista disponibilidad de la habitación solicitada para las fechas seleccionadas.

Luego de una reserva satisfactoria, el sistema debe proporcionar un comprobante de reserva, el cual podrá ser mostrado en pantalla como una confirmación o pre-factura, incluyendo:

- Nombre del huésped.
- Tipo de habitación reservada.
- Numero de la habitación reservada.
- Fechas de ingreso y salida.
- Número de noches reservadas.
- Costo total estimado de la estancia.

Registrar Salida (Check-Out)

El programa solo permitirá realizar el registro de salida de huéspedes que hayan sido previamente registrados y que tengan una estancia activa en el hotel.

El equipo de desarrollo deberá implementar un procedimiento para:

- Calcular el tiempo de estancia del huésped (en noches).

- Realizar el cobro correspondiente basado en el tipo de habitación reservada y el número de noches de hospedaje.

Valores de cobro:

- **Habitación Estándar:** \$120,000 por noche.
- **Suite:** \$250,000 por noche.

Procedimiento de Facturación:

1. Se multiplica el número de noches hospedadas por el valor de la habitación correspondiente.
2. Si un huésped registra salida antes de completar una noche de estancia, se cobrará igualmente el valor de una noche completa (no se realizan cobros parciales).

Condición de Pago Mínimo:

- El mínimo a pagar por una estancia, sin importar el tiempo, es el valor de una noche completa de la habitación reservada.

Generación de Factura:

Después de calcular el total a pagar, el sistema debe generar una factura en pantalla, que incluya:

- Nombre completo del huésped.
- Documento de identidad.
- Tipo y número de habitación.
- Fechas de ingreso y salida.
- Número total de noches.
- Costo total.

Administrador:

El programa solo permitirá acceder al módulo de administración a personas autorizadas que posean un usuario y contraseña de administración válidos.

Estos usuarios y contraseñas deben estar almacenados en un archivo de texto seguro que será leído por el sistema y cargado en un dataframe.

Si el usuario y la contraseña coinciden correctamente, el sistema permitirá el acceso al submenú de administración.

El módulo de administración deberá ofrecer los siguientes reportes:

- Total, de huéspedes registrados.
- Total, de habitaciones ocupadas actualmente.
- Total, de habitaciones disponibles.
- Total, de ingresos generados por reservas y estancias.
- Tiempo promedio de estancia por huésped.
- Lista de todos los huéspedes con historial de reservas.
- Huésped con la mayor cantidad de noches hospedadas.
- Huésped con la menor cantidad de noches hospedadas.

Visualización de Reportes Gráficos con Matplotlib

El sistema debe generar y mostrar al menos 8 gráficos diferentes con la librería

matplotlib:

1. Gráfica de barras
 - a. Comparación entre el número de habitaciones estándar y suites actualmente ocupadas.
2. Gráfica circular (pie chart)
 - a. Distribución porcentual de habitaciones ocupadas vs. disponibles.
3. Gráfica de líneas
 - a. Evolución del número de huéspedes que han hecho check-out por día.
4. Gráfica de barras horizontal
 - a. Comparación del número de noches hospedadas por cada huésped (Top 10).
5. Gráfica de dispersión (scatter plot)
 - a. Relación entre el número de noches y el valor total pagado por huésped.
6. Gráfica de pastel
 - a. Porcentaje de ingresos generados por tipo de habitación (estándar vs suite).
7. Histograma
 - a. Distribución de la duración de las estancias (en número de noches).
8. Gráfica combinada (barras + líneas)
 - a. Ingresos diarios (barras) y cantidad de huéspedes por día (línea), superpuestas.

Importante:

El profesor asumirá el rol de Product Owner (PO) en el marco de un desarrollo ágil, por lo cual el equipo de trabajo deberá entrevistarle y realizar preguntas de levantamiento de requisitos para aclarar dudas y definir mejor las funcionalidades y prioridades del sistema.

Conversación inicial con el profesor: ¡Haciendo las preguntas correctas!

Antes de comenzar a programar el sistema del Hotel BuenDescanso, es importante entender bien qué se espera del programa. Para eso, pueden tener una conversación informal con el profesor (que aquí actúa como si fuera el “cliente del sistema”) y hacerle preguntas como estas. No se preocupen si no saben programar todo todavía, lo importante es que entiendan la lógica del problema y las necesidades del usuario.

Objetivos y Propósito

- ¿Cuál es el objetivo principal de este programa para el hotel?
- ¿Qué problema estamos resolviendo con este sistema?
- ¿Quién lo va a usar? (¿el recepcionista?, ¿el administrador?, ¿el huésped?)
- ¿Qué cosas necesita hacer esa persona desde el sistema?

Funcionalidades y Requisitos

- ¿Qué cosas obligatoriamente debe hacer el sistema? (por ejemplo: registrar huésped, reservar habitación, hacer check-in, check-out, etc.)
- ¿Hay cosas que sería chévere tener, pero que no son urgentes ni necesarias?
- ¿Qué cosas el sistema no va a hacer?

Prioridades y Alcance

- Si el tiempo fuera poco, ¿cuáles son las funciones más importantes que sí o sí debe tener?
- ¿Qué funciones se podrían hacer después si hay tiempo?
- ¿Hay alguna versión futura del programa que se piensa mejorar más adelante?

Limitaciones y Restricciones

- ¿Podemos usar cualquier herramienta o lenguaje? ¿Solo Python?
- ¿Tenemos un tiempo límite para entregar el proyecto?
- ¿Hay reglas o condiciones especiales que debemos tener en cuenta?

Interfaz y Experiencia del Usuario

- ¿Ya hay algún dibujo o idea de cómo debe verse el programa en consola?
- ¿Cómo debe sentirse la persona que lo usa? ¿Fácil? ¿Claro? ¿Rápido?
- ¿El programa debe funcionar para personas con alguna discapacidad? (Por ejemplo: personas con dificultad visual)

Pruebas y Calidad

- ¿Cómo sabremos si el sistema funciona bien?
- ¿Qué cosas debe cumplir para decir que está bien hecho?
- ¿Quién revisará si todo está funcionando correctamente?

Riesgos y Mitigación

- ¿Qué podría salir mal durante el proyecto? (Por ejemplo: perder archivos, que no se entienda el problema, que se acabe el tiempo...)
- ¿Cómo podemos prepararnos para que eso no pase?
- ¿Qué hacemos si en el último momento algo falla?

Colaboración y Comunicación

- ¿Cómo nos vamos a comunicar con el profesor y entre nosotros?

- ¿Dónde vamos a guardar el proyecto? (Ej: en GitHub)
- ¿Quién se encarga de subir los archivos? ¿Quién organiza las tareas?

Registro de Actividades (Log de Eventos)

El programa debe guardar un archivo de texto donde quede registrado todo lo que hace el sistema, como si fuera una bitácora. Ese archivo debe incluir:

- Fecha y hora exacta (hasta milisegundos).
- Qué acción se hizo (ej: “Registrar huésped”, “Generar factura”, “Crear reserva”...).
- Cuánto tiempo tardó en ejecutarse.
- Separar los datos con tabulaciones o el símbolo |.
- Además, al principio del archivo debe salir información como:
- El nombre del usuario que ejecuta el programa.
- El sistema operativo (Windows, Linux, etc.).
- El tipo de equipo (plataforma).

Entregables

El trabajo debe ser entregado en un repositorio de GitHub bien organizado. Allí se debe incluir toda la información como si fuera un documento profesional, pero escrito en lenguaje Markdown, usando títulos, subtítulos, listas, imágenes, etc.

A continuación, se explican los entregables

El repositorio debe tener los siguientes procesos debidamente reportados.

1. Integrantes:

- a. En el archivo principal del repositorio (README.md), deben escribir los nombres de los integrantes del equipo, y sus roles.

Por ejemplo:

- Juan Pérez – Encargado del código.
- María López – Encargada de documentación.
- Kevin Restrepo – Encargado de pruebas.

2. Vínculos académicos y descripción:

- a. Cada integrante debe escribir a qué programa académico pertenece y qué habilidades puede aportar al proyecto.

Ejemplo:

- Soy estudiante de Ingeniería Ambiental. No tengo experiencia en programación, pero me esfuerzo por organizar bien la información y revisar que el sistema funcione paso a paso.

3. Nombre del proyecto y detalles:

El grupo debe inventar un nombre propio para su hotel (por ejemplo: Hotel SolNaciente, Hotel DescansoReal, etc.).

En el repositorio debe ir una imagen representativa del nombre del hotel (no

una foto del grupo), y una pequeña descripción del proyecto (máximo 50 palabras).

4. Licencia del software:

Se debe ingresar a esta página: <https://chooser-beta.creativecommons.org/> y elegir una licencia para el proyecto (como si fuera un trabajo creativo).

Luego, deben copiar el nombre de la licencia seleccionada y pegarla en el README.md.

5. Reporte de visión:

Escribir de forma sencilla para qué sirve el programa, qué problemas ayuda a resolver, y cómo sería útil si funcionara bien.

Ejemplo:

- Este software busca ayudar a un hotel a gestionar sus reservas, ingresos y salidas de huéspedes, generando reportes claros y organizados.

6. Especificación de requisitos:

En esta parte deben escribir qué debe hacer el programa (acciones concretas), explicado con sus propias palabras.

No se requiere usar palabras como “funcional” o “no funcional”. Basta con responder preguntas como:

- ¿Qué puede hacer el sistema?
- ¿Qué se espera que haga bien?
- ¿Qué debe evitarse o no debe hacer?
- ¿Qué librerías se requieren para llevar a cabo la solución?
- ¿Qué funciones se deben crear para resolver el problema planteado?
- ¿Qué estructuras de datos se van a implementar en la solución?

Se puede usar una tabla, una lista o párrafos. Lo importante es que quede claro lo que se espera del sistema desde el punto de vista de los usuarios (como el recepcionista o el administrador).

7. Plan de proyecto:

Describe las actividades, el cronograma (Diagrama de Gantt) y el presupuesto del proyecto.

- a. El presupuesto debe tener en cuenta que no se pagará en dinero sino en tiempo de práctica de formación. Es decir, el grupo del trabajo final lo componen tres estudiantes e invierten un total de 50 horas, estas serán pagadas a valor de práctica profesional. 1 SMLV.

8. Plan de versionado:

Describe las versiones del software y su avance cada que se realice un procedimiento relevante en días desde el inicio hasta la entrega final.

9. Algoritmo:

Una carpeta en el repositorio en GitHub con todos los archivos y código utilizados en el proyecto, esta carpeta debe ser llamada “src”.

10. Manual de usuario:

En una carpeta llamada doc, deben escribir un documento (puede ser en Markdown o Word convertido a PDF) donde expliquen cómo usar el

programa, paso a paso.

Ejemplo:

- Cómo iniciar el programa.
- Qué opciones se pueden seleccionar.
- Qué hacer para registrar un huésped.
- Cómo realizar un check-in o check-out, etc.

11. GitHub:

El líder del grupo debe usar su correo institucional de la UdeA para crear el repositorio en GitHub.

Debe invitar a sus compañeros y subir allí todos los archivos del proyecto.

Este proceso será explicado por el profesor en clase.

12. Sustentaciones:

Se deberá sustentar el software en la semana 16, en donde todos los integrantes del equipo tendrán cinco minutos para responder las preguntas del profesor. Esta actividad no es una exposición, es una discusión en donde se deberá tener el software disponible y disponible a ejecutarse.

13. Fechas y Documentos:

- a. Entrega 1: Para la primera entrega se debe enviar los puntos 1 a 7 Semana 10.
- b. Entrega 2: Todo lo descrito en el presente documento. Semana 16