**Proyecto Final Curso Programación Básica SC-115:**

**DreamWorld Casino**

**Grupo #8**

Max Hernández Medrano.

Jose David Jiménez Corrales.

Luis Felipe Cabalceta Barrantes.

**Universidad Fidélitas**

**Ingeniería en Sistemas de Computación**

Profesor:

**Msc. Álvaro Camacho Mora**

San José, Costa Rica

Julio, 2023

Tabla de contenido

[1 Introducción: 3](#_Toc143624547)

[2 Antecedentes: 4](#_Toc143624548)

[3 Objetivos: 5](#_Toc143624549)

[3.1 Objetivo General 5](#_Toc143624550)

[3.2 Objetivos Específicos 5](#_Toc143624551)

[4 Descripción de los Módulos Implementados 6](#_Toc143624552)

[4.1 Problema: 8](#_Toc143624553)

[4.2 Algoritmo y Estructuras de Software Requeridas. 9](#_Toc143624554)

[4.3 Bibliotecas/ Estructuras de Software Adicionales 12](#_Toc143624555)

[5 Requisitos Técnicos del Sistema: 12](#_Toc143624556)

[6 Manual del usuario 12](#_Toc143624557)

[7 Funciones: 13](#_Toc143624558)

[8 Bibliografía 15](#_Toc143624559)

# Introducción:

Introducción El presente trabajo busca establecer los parámetros y fundamentos lógicos para el desarrollo del proyecto final del curso Programación Básica, implementando las habilidades aprendidas a través del curso. En este documento se presenta el diseño de un software de última generación para el manejo del casino en línea “DreamWorld Casino”. El programa por desarrollar tiene la finalidad de convertirse en un software que ofrezca a las personas una experiencia de un casino innovador y amigable, pero que a la vez cumpla con todos los parámetros de seguridad requeridos para transmitir la confianza que buscan los usuarios. El software tiene como objetivo recrear la experiencia de un casino real de alta gama, proporcionando medidas de seguridad sólidas, opciones de registro y acceso avanzado para la configuración del sistema. A continuación, se detallan los módulos y funcionalidades clave del software.

# Antecedentes:

Cuando pensamos en casinos, a menudo pensamos en la ciudad de Las Vegas en los Estados Unidos. Sin embargo, la historia de los casinos se remonta al año 2300 a.C., donde se han encontrado evidencias de los primeros tipos de apuestas (Casino - Wikipedia, s.f.) ¹. Posteriormente, el primer casino fue creado en Italia en el año 1600 y no fue hasta el año 1930 que se popularizó en los Estados Unidos al legalizar las apuestas y crear centros de apuestas (casinos) en ciudades como Las Vegas, Nevada y Atlantic City, Nueva Jersey (Casino - Wikipedia, s.f.) ¹.

Uno de los juegos más populares en los casinos son las tragamonedas. Este juego es ideal ya que ofrece una satisfacción inmediata para el usuario que gana e incita a seguir apostando. Las primeras tragamonedas fueron inventadas en el Viejo Oeste a mediados de los años 1800 y no fue hasta finales de los años 1940 que se popularizaron en Las Vegas debido a que su aparición propició que los hoteles/casinos obtuvieran grandes ganancias en poco tiempo. Su evolución ha sido marcada principalmente por la digitalización de los mecanismos que controlan sus sistemas hasta llegar a nuestros días, donde tenemos tragamonedas en muchas partes de las ciudades y no solo en casinos o sitios de apuestas (Slot machine - Wikipedia, s.f.) ⁵.

Otro juego de casino por excelencia es el Blackjack o 21. Este juego consiste en que un jugador logre obtener la cifra de 21 al sumar el valor de sus cartas y con esto, derrotar a la banca (crupier). Sus orígenes se remontan al siglo XVII en los casinos franceses y su popularidad en América se remonta al viaje de inmigrantes franceses en el siglo XIX. Actualmente es un juego popular que es ampliamente jugado por cualquier persona sin importar su edad debido a su facilidad de entendimiento, velocidad y generación de grado de satisfacción (Blackjack - Wikipedia, s.f.) ⁹.

En resumen, la historia de los casinos y sus juegos más populares como las tragamonedas y el Blackjack se remonta a muchos siglos atrás y ha evolucionado a lo largo del tiempo para adaptarse a las nuevas tecnologías y necesidades de los jugadores.

# Objetivos:

## Objetivo General

Desarrollar un programa computacional en lenguaje Python que explote las habilidades de programación aprendidas y responda a los requerimientos establecidos por el cliente.

## Objetivos Específicos

▪ Demostrar el entendimiento de los módulos solicitados a través de la

generación de la identificación del problema, estructuras de datos requeridos y

algoritmo necesario para la resolución del problema.

▪ Diseñar un programa que logre la cohesión de submódulos de una forma lógica

para resolver los requerimientos de software y que sea accesible además de seguro.

▪ Defender el diseño implementado utilizando las correctas habilidades de

comunicación para demostrar la resolución correcta del problema planteado.

# Descripción de los Módulos Implementados

Módulo de Usuarios:

La información de los usuarios se almacena en un archivo usuarios.txt.

Se ha implementado la clase Usuario para gestionar la información de cada usuario, incluyendo su ID, nombre, PIN y saldo.

Funciones de Utilidad:

Varias funciones de utilidad se han creado para manejar la entrada de datos y mensajes para una interacción amigable con el usuario.

Módulo de Registro de Usuarios:

Los nuevos usuarios pueden registrarse a través del proceso de registro que incluye la creación de un ID alfanumérico, solicitud de nombre, elección de un PIN, y realización de un depósito mínimo.

La información de los usuarios se almacena en el archivo usuarios.txt y se crea una instancia de la clase Usuario para cada usuario registrado.

Módulo del Casino:

Los usuarios registrados pueden acceder al casino después de autenticarse mediante su ID y PIN.

Se muestra un menú con diversas opciones, como retirar y depositar dinero, ver el saldo actual, jugar juegos en línea y eliminar su cuenta.

Módulo de Juegos en Línea:

Se ofrecen dos juegos en línea: Blackjack y Tragamonedas.

Los juegos en línea incluyen funciones para mostrar instrucciones, gestionar apuestas, ejecutar rondas de juego y determinar los resultados.

Módulo de Configuración Avanzada:

Los usuarios autenticados pueden acceder a la configuración avanzada, donde se pueden modificar valores como tipos de cambio y valores de apuesta.

Funciones de Carga y Guardado:

Funciones de carga y guardado de información de usuarios se han implementado para administrar la información almacenada en el archivo usuarios.txt.

Función Principal (main):

La función principal maneja la interacción con el usuario, mostrando el menú principal y dirigiendo a los usuarios a los diferentes módulos según su elección.

Esta estructura modular permite una fácil expansión y mantenimiento del software, asegurando un alto nivel de eficiencia y usabilidad para los usuarios del casino en línea DreamWorld Casino. Cada módulo ha sido desarrollado de manera independiente para garantizar la coherencia y facilidad de desarrollo, permitiendo así un funcionamiento robusto y confiable del software

## Problema:

En este proyecto se plantea la necesidad de diseñar y desarrollar un software de última generación para el manejo del casino en línea "DreamWorld Casino". El casino busca recrear la experiencia de un casino real de alta gama en un entorno en línea, proporcionando medidas sólidas de seguridad, opciones avanzadas de registro y acceso, así como funcionalidades de juego para los usuarios. El objetivo es ofrecer una plataforma interactiva y atractiva que permita a los usuarios disfrutar de juegos como Blackjack y Tragamonedas, realizar transacciones de dinero de manera segura, acceder a configuraciones avanzadas y gestionar su perfil de usuario.

El proyecto también implica la creación de un sistema eficiente de manejo de usuarios y datos, donde los usuarios pueden registrarse, autenticarse, realizar depósitos y retiros de dinero, jugar a juegos de casino y realizar configuraciones personalizadas. Además, el software debe asegurar la integridad de los datos y garantizar una experiencia de usuario satisfactoria y fluida en dispositivos de gama media.

En resumen, el problema a resolver en este proyecto es el diseño y desarrollo de un software completo y funcional que permita la operación exitosa de un casino en línea, abordando desafíos como la seguridad de los datos, la experiencia de usuario, la implementación de juegos de casino y la gestión de transacciones financieras.

## Algoritmo y Estructuras de Software Requeridas.

Algoritmo:

* Mostrar mensaje de bienvenida y menú principal.
* Esperar la selección del usuario.

Mientras el usuario no seleccione "Salir":

Si el usuario selecciona "Registro de usuario nuevo":

* Realizar el proceso de registro de usuario nuevo:

a. Solicitar al usuario que elija un ID alfanumérico de al menos cinco caracteres.

b. Verificar la disponibilidad del ID en los registros.

c. Si el ID no está disponible:

- Mostrar mensaje de error.

- Permitir hasta tres intentos de ingreso de un ID válido.

d. Solicitar al usuario su nombre.

e. Solicitar al usuario que elija un PIN numérico de al menos seis dígitos.

f. Confirmar el PIN ingresado.

g. Si los PIN no coinciden:

- Mostrar mensaje de error.

- Permitir hasta tres intentos de ingreso de un PIN válido.

h. Solicitar al usuario realizar un depósito mínimo en dólares, colones o bitcoin.

i. Verificar si el depósito es válido y suficiente.

j. Si el depósito no es válido o suficiente:

- Mostrar mensaje de error.

- Permitir hasta tres intentos para realizar el depósito mínimo.

k. Guardar la información del nuevo usuario en archivos.

l. Mostrar mensaje de registro exitoso.

m. Volver al menú principal.

Si el usuario selecciona "DreamWorld Casino":

* Realizar el proceso de inicio de sesión en el casino:

a. Cargar información de usuarios registrados desde archivos.

b. Solicitar al usuario su ID y PIN para autenticación.

c. Verificar la autenticación del usuario.

d. Si la autenticación falla:

- Mostrar mensaje de error.

- Permitir hasta tres intentos de autenticación.

e. Mostrar el menú de opciones del casino.

f. Esperar la selección del usuario.

Si el usuario selecciona "Retirar dinero":

- Realizar el proceso de retiro de dinero.

Si el usuario selecciona "Depositar dinero":

- Realizar el proceso de depósito de dinero.

Si el usuario selecciona "Ver saldo actual":

- Mostrar el saldo actual del usuario.

Si el usuario selecciona "Juegos en línea":

* Realizar el proceso de juego en línea:

a. Mostrar opciones de juegos (Blackjack y Tragamonedas).

b. Esperar la selección del usuario.

Si el usuario selecciona "Blackjack":

- Realizar el proceso de juego de Blackjack.

Si el usuario selecciona "Tragamonedas":

- Realizar el proceso de juego de Tragamonedas.

Si el usuario selecciona "Eliminar usuario":

- Realizar el proceso de eliminación de usuario.

g. Volver al menú principal.

Si el usuario selecciona "Configuración Avanzada":

* Realizar el proceso de configuración avanzada:

a. Solicitar autenticación con PIN especial.

b. Mostrar opciones de configuración.

c. Esperar la selección del usuario.

Si el usuario selecciona "Modificación de valores":

- Permitir al usuario modificar configuraciones.

d. Volver al menú principal.

* Volver al menú principal.
* Mostrar mensaje de despedida.
* Finalizar el programa.

Estructura de Software:

Importación de Bibliotecas:

import getpass, random, os, time: Estos comandos importan las bibliotecas getpass, random, os y time, que proporcionan funciones y utilidades específicas. getpass se utiliza para funciones seguras, random para generar números aleatorios, os para interactuar con el sistema operativo y archivos, y time para gestionar pausas y medir el tiempo.

Variables Globales:

usuarios = {}: Crea un diccionario vacío llamado usuarios para almacenar la información de los usuarios registrados. Las claves del diccionario serán los nombres de usuario y los valores serán sus contraseñas y saldos.

saldos = {}: Crea otro diccionario vacío llamado saldos para almacenar los saldos de los usuarios.

moneda = "$": Asigna el símbolo "$" a la variable moneda, que se utiliza para representar la moneda utilizada en el casino.

monto\_minimo = 10: Establece el monto mínimo requerido para jugar como $10.

Funciones de Registro y Depósito:

def registrarUsuario(): Define una función para el registro de usuarios. Dentro de la función, se solicita al usuario su nombre, se verifica su disponibilidad, y se crea una contraseña segura usando hashlib.

def depositoInicial(): Define una función para el depósito inicial de los usuarios. El usuario ingresa su nombre y el monto que desea depositar. El saldo del usuario se actualiza en el diccionario saldos.

Funciones de Autenticación y Acceso al Casino:

def autenticarUsuario(): Pide al usuario su nombre y contraseña, luego verifica si coinciden con los registros almacenados en el diccionario usuarios. Si es así, el usuario se autentica.

def jugarBlackjack(): Inicia un juego de Blackjack. El usuario juega contra la computadora, se generan cartas aleatorias y se aplican las reglas del juego.

def jugarTragamonedas(): Simula una máquina tragamonedas donde se generan números aleatorios y se comparan para determinar si el usuario gana o pierde.

Funciones para Mostrar Saldo y Eliminar Usuario:

def mostrarSaldo(usuario): Toma el nombre de usuario como entrada y muestra su saldo correspondiente desde el diccionario saldos.

Nota: El código proporcionado no incluye una función para eliminar usuarios, aunque podría implementarse de manera similar a la función de mostrar saldo.

Función de Configuración Avanzada:

def configuracionAvanzada(): Define una función para acceder a la configuración avanzada. Aquí es donde podrían implementarse comandos para eliminar usuarios y modificar otros valores del sistema.

Función del Menú Principal:

def mostrarMenu(): Muestra las opciones del menú principal al usuario y recibe su elección. Luego, redirige al usuario a la función correspondiente según su elección.

En resumen, estos comandos y funciones se combinan para crear una estructura cohesiva que permite a los usuarios registrar cuentas, autenticarse, acceder a diferentes juegos, realizar depósitos, verificar saldos y más, todo mientras mantienen una organización clara y modular del código.

## Bibliotecas/ Estructuras de Software Adicionales

getpass: Utilizada para ingresar contraseñas de manera segura, ocultando la entrada en la consola.

os: Utilizada para operaciones relacionadas con el sistema operativo, como manejo de archivos.

Random: para generar números aleatorios,

Time: para gestionar pausas y medir el tiempo.

# Requisitos Técnicos del Sistema:

Python 3.x: El sistema debe tener instalado Python 3.x para ejecutar el programa.

Bibliotecas: Las siguientes bibliotecas deben estar instaladas:

Getpass: Para el almacenamiento seguro de contraseñas.

random: Para generar números aleatorios en los juegos.

time: Para introducir pausas realistas en el juego.

# Manual del usuario

Ejecución del Programa:

* Abre tu terminal o línea de comandos.
* Navega hasta la ubicación donde tienes el archivo casino.py.
* Ejecuta el programa con el comando: python casino.py.
* Registro de Usuario:
* Al iniciar el programa, verás el menú principal.
* Selecciona la opción "Registrarse" e ingresa tu nombre de usuario y contraseña.
* Tu contraseña será almacenada de forma segura.
* Depósito Inicial:
* Después de registrarte, puedes realizar un depósito inicial.
* Ingresa la cantidad de dinero que deseas depositar en tu cuenta.
* Acceso al Casino:
* Una vez que hayas registrado y depositado dinero, puedes acceder al casino.
* Elige entre las opciones "Blackjack" o "Máquina Tragamonedas".
* Juego de Blackjack:
* Si eliges jugar al Blackjack, sigue las instrucciones en pantalla para jugar.
* Ingresa la cantidad que deseas apostar y sigue las reglas del juego.
* Máquina Tragamonedas:
* Si decides jugar a la Máquina Tragamonedas, sigue las indicaciones en pantalla.
* Ingresa la cantidad a apostar y gira los rodillos para obtener combinaciones.
* Ver Saldo:
* En cualquier momento, puedes verificar tu saldo seleccionando "Ver Saldo" en el menú.
* Eliminar Usuario:
* En el menú principal, la opción "Configuración Avanzada" te permite eliminar tu cuenta.
* Salir del Programa:
* Si deseas salir del programa, selecciona la opción "Salir" en el menú principal.

# Funciones:

“Una función es un bloque de código que solo se ejecuta cuando es llamado”. ***(1)***

Según esa definición, las funciones aplicadas a la lógica detrás de nuestro casino, buscan simplificar el código en momentos cuando se necesite llamar un bloque de código para ejecutarlo, como el caso del registro de usuario y mostrar menú, además de las secciones que posteriormente se desarrollarán.

**Def():**

“Una función en Python es definida a través de la keyword def()”. (***2)*** ***(Python Functions, n.d.).***

nuevoUsuario():

Esta función maneja el proceso de registro de un nuevo usuario en el casino en línea.

Propósito: Permite al usuario elegir un ID alfanumérico, un PIN de seguridad y realizar un depósito mínimo para completar el registro. La función valida la entrada y almacena la información del usuario en un archivo.

depositoMinimo():

Esta función guía al usuario a realizar un depósito mínimo para completar su registro.

Propósito: Solicita al usuario el monto y la moneda en la que desea realizar el depósito, calcula el equivalente en dólares si es necesario y verifica si el depósito cumple con el monto mínimo requerido.

retirarDinero(idUsuario):

Esta función está en desarrollo y se espera que maneje el proceso de retirar dinero de la cuenta del usuario.

Propósito: Permitir al usuario retirar una cantidad específica de dinero de su saldo en la cuenta. Aún no se ha implementado la lógica completa para esta función.

depositarDinero(idUsuario):

Esta función está en desarrollo y se espera que maneje el proceso de depositar dinero en la cuenta del usuario.

Propósito: Permitir al usuario depositar una cantidad específica de dinero en su saldo de cuenta. Aún no se ha implementado la lógica completa para esta función.

verSaldo(idUsuario):

Esta función está en desarrollo y se espera que muestre el saldo actual de la cuenta del usuario.

Propósito: Mostrar al usuario el saldo actual en su cuenta, permitiéndole verificar cuánto dinero tiene disponible. Aún no se ha implementado la lógica completa para esta función.

juegosEnLinea(id\_usuario):

Esta función está en desarrollo y se espera que permita al usuario jugar juegos en línea dentro del casino.

Propósito: Ofrecer al usuario la posibilidad de jugar diferentes juegos en línea dentro del casino, como Blackjack o Tragamonedas. Aún no se ha implementado la lógica completa para esta función.

eliminarUsuario(id\_usuario):

Esta función está en desarrollo y se espera que maneje el proceso de eliminación de un usuario y sus registros.

Propósito: Permitir al usuario eliminar su cuenta y registros asociados del casino en línea. Aún no se ha implementado la lógica completa para esta función.

configuracionAvanzada():

Esta función muestra un mensaje de desarrollo y se espera que maneje configuraciones avanzadas.

Propósito: Brindar al usuario acceso a configuraciones avanzadas del sistema, como tipos de cambio y valores de apuesta. Aún no se ha implementado la lógica completa para esta función.

salir():

Esta función imprime un mensaje de despedida al usuario.

Propósito: Finalizar el programa de manera amigable, agradeciendo al usuario por utilizar el casino en línea.

mostrarMenu():

Esta función muestra el menú principal del programa.

Propósito: Presentar al usuario las opciones disponibles, como registrar un nuevo usuario, acceder al casino, configuración avanzada o salir del programa. Captura la elección del usuario y ejecuta la función correspondiente.

Conclusión:

El proyecto del casino en línea proporciona una base para comprender los conceptos básicos de programación, manipulación de datos y diseño de software. A través del análisis del código y sus características, podemos extraer las siguientes conclusiones:

* Aprendizaje de Python: El proyecto es adecuado para principiantes que deseen aprender a programar en Python. Utiliza estructuras de control como bucles, condicionales y funciones, lo que ayuda a familiarizarse con la sintaxis y las características del lenguaje.
* Gestión de Usuarios: El sistema muestra cómo registrar usuarios, autenticarlos y administrar sus saldos. Esto es fundamental para muchas aplicaciones que requieren autenticación y perfiles de usuarios.
* Interacción con el Usuario: El proyecto ilustra cómo interactuar con el usuario a través de la entrada y salida de texto en la consola. Sin embargo, en aplicaciones más avanzadas, se requerirían interfaces de usuario más amigables, posiblemente utilizando bibliotecas gráficas.
* Seguridad Básica: La utilización de la biblioteca hashlib para almacenar contraseñas de manera segura es un buen punto de partida en términos de seguridad. No obstante, en entornos de producción, se requerirían medidas de seguridad más avanzadas, como el uso de bases de datos seguras y técnicas de autenticación más robustas.
* Mejoras Posibles: El proyecto podría mejorarse agregando más juegos de casino, ampliando las opciones de configuración, implementando un sistema de registro más completo con validación de datos y explorando formas de hacer el programa más eficiente y modular.

# Bibliografía

*Python Functions*. (n.d.). https://www.w3schools.com/python/python\_functions.asp

*Python len() Function*. (n.d.). <https://www.w3schools.com/python/ref_func_len.asp>

 ¹ Casino - Wikipedia. (s.f.). Recuperado el 20/8/2023, desde <https://en.wikipedia.org/wiki/Casino>.

⁵ Slot machine - Wikipedia. (s.f.). Recuperado el 20/8/2023, desde <https://en.wikipedia.org/wiki/Slot_machine>.

⁹ Blackjack - Wikipedia. (s.f.). Recuperado el 20/8/2023, desde <https://en.wikipedia.org/wiki/Blackjack>.

global: https://www.w3schools.com/python/python\_variables\_global.asp

append: https://www.w3schools.com/python/ref\_list\_append.asp

Manejo de Errores: https://docs.python.org/3/tutorial/errors.html

Diccionarios: https://www.w3schools.com/python/python\_dictionaries.asp

Strip: https://www.w3schools.com/python/ref\_string\_strip.asp

Split: https://www.w3schools.com/python/ref\_string\_split.asp

Return: https://www.w3schools.com/python/ref\_keyword\_return.asp

Slicing: https://www.w3schools.com/python/python\_strings\_slicing.asp

Pop: https://www.w3schools.com/python/ref\_list\_pop.asp