| ***UTN – FRMDP Mar del Plata TUP - Programación 1***  ***Trabajo Práctico Final***  *Junio 2024* | ***Integrantes del grupo*** | ***Nota*** |
| --- | --- | --- |

# Introducción

Con el propósito principal de integrar todo lo aprendido en la materia Programación 1 hemos planteado la siguiente problemática:

* Codificar un sistema Bancario, encargado por el banco BF S.A. (Bicicleta Fácil S.A.) para el control de Clientes y sus Movimientos de dinero
* Generar la persistencia de los Clientes y sus Movimientos

# Fundamentación

El valor pedagógico de la propuesta se apoya en el aprendizaje colaborativo (se formarán grupos de 4 alumnos) a partir del desarrollo de un proyecto de software. Para que este tipo de proyectos sea más exitoso, deben llevarse a cabo desde un enfoque que facilite alcanzar los Objetivos de Aprendizaje propuestos.

Una de las ideas centrales es desarrollar competencias profesionales y preparar al futuro programador para el mundo laboral y el trabajo en equipo.

En un ambiente de aprendizaje colaborativo, los estudiantes:

* Construyen conocimiento y en lugar de recibirlos en forma pasiva;
* Se involucran y se comprometen directamente con el descubrimiento de nuevo conocimiento;
* Se exponen a puntos de vista alternativos e ideas contrapuestas, de forma tal que pueden sacar sus propias conclusiones y así transformar conocimientos y experiencias previas y de esta manera comprender con mayor profundidad;
* Transfieren conocimientos y habilidades a nuevas situaciones o circunstancias;
* Se responsabilizan y apropian tanto de su aprendizaje continuo de contenidos curriculares, como del desarrollo propio de competencias;
* Los estudiantes colaboran para el aprendizaje del grupo y el grupo colabora en el aprendizaje individual de estos.

# Objetivos

De aprendizaje:

* Incorporar Arreglos, Matrices, Modularización, Estructuras de Datos y Archivos Binarios.
* Trabajar en forma colaborativa.

Metodológicos:

* Ser capaces de trabajar en un proyecto complejo, aplicando técnicas de desarrollo de software.
* Lograr integrar contenidos de otras asignaturas.
* El grupo deberá ir mostrando el avance sobre el trabajo en clase.

**Esquema de las Estructuras**

## Estructura de Domicilio: stDomicilio

## char calle[30];

## char nro[6];

## char localidad[50];

## char provincia[40];

## char cpos[6];

## Estructura de Clientes: stCliente

int id; /// campo único y autoincremental

int nroCliente;

char nombre[30];

char apellido[30];

char dni[10];

char email[30];

stDomicilio domicilio;

char telefono[12];

int eliminado; /// 0 si está activo - 1 si está eliminado

**Estructura de Cuentas: stCuenta**

## int id; /// campo único y autoincremental

int idCliente /// Id del Cliente dueño de la Cuenta;

int nroCuenta;

int tipoDeCuenta; /// 1. Caja de Ahorro en Pesos, 2. Caja de Ahorro en Dólares, 3. Cta Cte en $

float costoMensual /// Costo de mantenimiento del producto;

float saldo; /// Debe actualizarse al modidificar o cargar un movimiento.

int eliminado; /// 0 si está activo - 1 si está eliminado

**Estructura de Movimientos: stMovimiento**

int id; /// campo único y autoincremental

int idCuenta; /// Id de la cuenta

char detalle[100];

float importe;

int anio;

int mes; /// 1 a 12

int dia; /// 1 a … dependiendo del mes

int eliminado; /// 0 si está activo - 1 si está eliminado

# Modo de Evaluación del Trabajo Práctico

* Se establece el desarrollo de un trabajo práctico final, brindando una fecha límite de entrega del mismo: **Según planificación de cada comisión**
* La aprobación del trabajo práctico estará sujeta a los puntajes considerados en la tabla debajo.
* Es obligatorio la presentación de este trabajo para aprobar la materia.

| ***Apartado*** | ***Puntaje*** | ***Obtenido*** |
| --- | --- | --- |
| Funciones para ABMCL de Clientes, Cuentas y Movimientos   * Alta, Baja, Modificación, Consulta y Listados de Clientes * Alta, Baja, Modificación, Consulta y Listados de Cuentas por Cliente * Alta, Baja, Modificación, Consulta y Listados de Movimientos por Cuenta y por mes. * Alta de movimientos aleatorios del año actual anteriores a la fecha actual. La función deberá agregar 1000 eligiendo de manera aleatoria el cliente el mes y el día. Si el importe es negativo en descripción deberá insertar un texto relativo a un pago, transferencia a cuenta propia, transferencia a cuenta de terceros, pago de TC, o pagos con TD. Si es positivo en descripción deberá insertar un texto relativo a un cobro, ej. crédito Haberes, transferencia de cuenta propia, transferencia de cuenta ajena, acreditación de plazo fijo, etc. **Cada vez que se ingresa o modifica un movimiento se debe actualizar el saldo de la cuenta.** * Validación en el ingreso de los Datos. * Buscar Movimientos por fecha. * Buscar Clientes. | 35 |  |
| Funciones para manejo de archivos binarios   * Persistencia de datos en Archivos. * Validar si el Clientes ya existe (para alta de Cliente). | 35 |  |

| Función main () y funciones de manejo de vistas del sistema Orden y prolijidad del código entregado. Explicación presencial del sistema   * Correcta modularización de las funciones. * Correcto uso de parámetros. * Prolijidad general del código. * Reutilización de las funciones. * Comentarios del código.   Como metodología de trabajo, se requiere crear una carpeta en Google Drive que será compartida a todos los miembros del grupo (y también al equipo docente, publicando el link vía campus virtual en el foro correspondiente), con el fin de plasmar los avances del proyecto de forma de construir la siguiente documentación a entregar:   * Diario de trabajo: Semana a semana qué actividades se desarrollaron y el responsable de cada una. * Matriz de soluciones: Que problema tuvieron y cómo lo resolvieron. * Diagrama de estructuras: Esquema de las estructuras utilizadas y sus relaciones. * Manual de usuario: Breve explicación de cómo funciona el sistema, pueden usar imágenes, videos, presentaciones, etc. | 30 |  |
| --- | --- | --- |

Al realizar la entrega final, deberán tener en cuenta los siguientes puntos:

* Documentación completa según lo requerido por la cátedra, que será subido al campus de la UTN.
* Código del sistema completo, compilado y sin errores
* Explicación presencial del sistema. Además, tiene que cumplir con las siguientes funcionalidades:
* Organizar en Menús
* Registrar Clientes y Movimientos
* Persistir datos en el archivo

**Tabla de puntuación:**

| **Obtenido** | 10 | 20 | 30 | 40 | 50 | **60** | **70** | **80** | **90** | **100** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nota** | 1 | 2 | 3 | 4 | 5 | **6** | **7** | **8** | **9** | **10** |
|  | Desaprobado | | | | | **Aprobado** | | | | |