**2do D - Primer Parcial**

| **Comisión** | **Docentes** | **Instancia de evaluación** | **Cuatrimestre** |
| --- | --- | --- | --- |
| 2do D - Lunes y Martes 18:30 a 22:30 | M. Cerizza - L. Galarza - E. Prieto | Primer Parcial | 2do 2021 |

**Requerimiento**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#requerimiento)

Somos desarrolladores de principios de la década del 2000 (pero con tecnología y conocimientos del futuro 😉) y nos contratan del ***cibercafé "El vicio"*** para desarrollar un sistema que facilite la administración de los servicios que ofrece el negocio.



El local cuenta con 10 **computadoras** y 5 **cabinas telefónicas**.

Tanto las computadoras como las cabinas cuentan con un **identificador** alfa-numérico, las primeras con el prefijo 'C' seguido de un número (ejemplo, 'C01', 'C02', etc) y las segundas con el prefijo 'T' (ejemplo, 'T01', 'T02', etc).

De los teléfonos también se debe saber:

* **Tipo** (A disco / Con teclado)
* **Marca**

De las computadoras también se debe saber:

* **Software instalado** (office, messenger, icq, ares).
* **Periféricos disponibles** (cámara, auriculares, micrófono).
* **Juegos disponibles** (Counter Strike, Diablo II, Mu Online, Lineage II, Warcraft III, Age of Empires II)
* **Especificaciones de hardware** (procesador, ram, placa de video, etc).
* Siéntanse libres de agregar otras opciones de software y juegos entre las existentes en la época.
* Cada computadora tiene que tener asociados distintos programas, juegos y periféricos, pero no todos ni repetirse (para simular una situación real).

El **costo de uso** en el caso de las llamadas telefónicas se mide por minuto y depende de la distancia:

* ***Local***: $1.00 por minuto. Se reconoce a una llamada como local cuando el prefijo contiene un '11' o '011' luego del código de país (54 - 11 - #### - ####).
* ***Larga distancia***: $2.50 por minuto. Se reconoce a una llamada como de larga distancia cuando el prefijo contiene un número distinto de '11' o '011' luego del código de país (54 - #### - #### - ####).
* ***Internacional***: $5.00 por minuto. Se reconoce a una llamada como de larga distancia cuando el código de país es distinto de '54'.

El **costo de uso** de las computadoras es $0.50 por cada **fracción mínima de 30 minutos**. Se redondeará siempre hacia arriba (por ejemplo, 35 minutos se consideran una hora de uso).

El **usuario** (también llamado *"el pibe que atiende el ciber"*) tiene que poder realizar las siguientes acciones sobre el sistema:

* Visualizar **clientes** a la espera de ser atendidos.
* Asignar una máquina o cabina a cada cliente en espera.
* Marcar como finalizado el uso de una máquina o cabina y cobrar por el servicio.
* Consultar **estadísticas históricas** del local.

**Consigna**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#consigna)

Desarrollar una aplicación de escritorio con *Windows Forms* que resuelva las siguientes necesidades del cliente.

Se deberá poder:

* Ver la **lista de clientes a la espera** de atención ordenados por orden de llegada.
  + Se deberá poder visualizar o consultar **dni**, **nombre**, **apellido** y la **edad** del cliente.
  + Se debe poder consultar qué software, hardware o juego necesita el cliente. Esta elección debe ser simulada de forma aleatoria al iniciar el programa.
* Contar con una **barra de información de la aplicación** donde figure el nombre del operador conectado (**que deberá ser TU nombre**) y la fecha actual (sin la hora).
* Visualizar fácilmente el **estado actual** de cabinas o computadoras (disponibles/en uso).
* Poder asignar una cabina o computadora al siguiente cliente en la fila.
  + De las computadoras deberá ver toda la información de hardware y software disponible, y **sólo** poder seleccionar una que contenga lo que el cliente necesita.
  + No se deben poder asignar cabinas o computadoras que se encuentren en uso.
  + Para asignar una computadora se deberá ingresar el **tiempo solicitado de alquiler** u opción *"Libre"* que dejará correr el tiempo hasta que el cliente pida finalizar. La mínima fracción es de 30 minutos, es decir sólo se puede alquilar en bloques de media hora (1/2, 1, 1 y 1/2, 2, etc).
  + Para asignar un teléfono se deberá cargar el teléfono completo al que se va a llamar. Diferenciar **código de país** y **prefijo de localidad** del resto del número.
* Poder indicar la finalización del uso de una computadora o teléfono.
  + Al finalizar se le deberá mostrar al usuario el tiempo de uso, el costo final bruto por el servicio y el neto a pagar una vez aplicado IVA.
  + No se debe poder marcar como finalizado algo que no está en uso.
* Poder visualizar estadísticas históricas:
  + Lista de computadoras ordenadas por minutos de uso de forma descendente.
  + Lista de cabinas ordenadas por minutos de uso de forma descendente.
  + Ganancias totales y clasificadas por servicio (teléfono/computadora).
  + Horas totales y la recaudación por tipo de llamada.
  + El software más pedido por los clientes.
  + El periférico más pedido por los clientes.
  + El juego más pedido por los clientes.

Por fines prácticos de la evaluación:

* Un segundo de la vida real será equivalente a un minuto del ciber.
* Un minuto de la vida real será equivalente a una hora del ciber.

**Criterios de evaluación**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#criterios-de-evaluaci%C3%B3n)

**Documentación**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#documentaci%C3%B3n)

Se deberá completar la información indicada en el archivo README.md que se encuentra en el repositorio.

Para trabajar con este archivo se deberá utilizar el lenguaje de marcado [Markdown](https://www.markdownguide.org/" \t "_blank).

**❗❗ No se corregirá ni revisará ningún parcial que no presente esta documentación completa.**

**Secciones e información a documentar**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#secciones-e-informaci%C3%B3n-a-documentar)

* **Título**: Ponerle un nombre a la aplicación.
* **Sobre mí**: Presentarse brevemente. Contar su experiencia programando y lo que significó para vos este trabajo (¿fue un desafío? ¿fue fácil? ¿aprendiste? ¿te divertiste? etc...).
* **Resumen**: Explicar qué hace la aplicación y cómo se usa a grandes rasgos.
* **Diagrama de clases**: Pegar una foto del diagrama de clases correspondiente a la lógica de negocio. Se debe construir con [la herramienta de Visual Studio](https://docs.microsoft.com/en-us/visualstudio/ide/class-designer/how-to-add-class-diagrams-to-projects?view=vs-2019) y deberá estar actualizado a la última versión entregada de la solución.
* **Justificación técnica**: Indicar tema a tema (de la clase 01 a la 10) dónde se fue aplicando en el código **y por qué se decidió implementarlo de esa forma**. Toda decisión tiene que estar argumentada con razones técnicas que giren alrededor de los pilares de la programación orientada a objetos y cuestiones de mantenibilidad, código limpio, flexibilidad al cambio, experiencia de usuario, accesibilidad, uso seguro, rendimiento y eficiencia.
  + Suma identificar pros y contras, si los tienen en mente.
  + El objetivo es que demuestren que saben lo que hacen y que tomaron decisiones con criterio y no mecanizadas.
  + Si se utilizó alguna biblioteca externa también se deberá justificar la elección.
* **Propuesta de valor agregado**: En esta sección se explicará y justificará la funcionalidad adicional propuesta para el [punto de promoción](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#condiciones-m%C3%ADnimas-para-promocionar).

**Condiciones mínimas de aprobación**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#condiciones-m%C3%ADnimas-de-aprobaci%C3%B3n)

Para alcanzar la aprobación (nota 4) se deberán cumplir todas las siguientes pautas:

* Respetar TODAS las [reglas de estilo de la cátedra](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/introduccion/guia-estilos) y buenas prácticas indicadas en clase. Se corregirán todas las pantallas (visualización) y calidad de código según las exigencias de la cursada.
* Compilar sin errores ni advertencias (sí se admiten sugerencias del IDE).
* Debe resolver TODAS las necesidades del cliente (planteadas en la consigna y en el requerimiento) y no tener errores en tiempo de ejecución.
* Separar de forma física (distintos proyectos) la capa de presentación (interfaz de usuario) de la lógica de negocio.
* Los datos de uso que no se carguen manualmente deben encontrarse pre-cargados (como literales) así como algunos datos históricos que simulen ejecuciones previas del programa ya que no contaremos con persistencia de datos.
* Todos los formularios deberán contar con un botón de ayuda que explique brevemente cómo se utilizan las funcionalidades.
* Para todos los formularios se debe poder maximizar, minimizar y cambiar el tamaño de la ventana.
  + La posición y/o tamaño de los controles deberá ajustarse con la ventana. Pista: hay una propiedad específica para esto, la vimos en clase.
  + Deberá existir un límite mínimo para ajustar el tamaño que será aquel donde se pierde visibilidad de los controles o se dificulta el trabajo con la aplicación.
* Las clases y sus miembros deberán estar correctamente documentados con la herramienta de documentación xml.
* Validar todos los ingresos de datos (cuando corresponda) mostrando mensajes claros para el usuario cuando un dato sea inválido.
* Abstraer las entidades y realizar un diseño orientado a objetos. Aplicar todos los temas de las clases 01 a la 10.
  + Al menos dos formularios.
  + Se valorará el uso justificado de formularios MDI, pantallas modales y no-modales.
  + Ninguna entidad se debe comparar por defecto, sino por uno o varios de sus atributos (idealmente por su identificador). Se deberá cambiar el comportamiento todos los métodos de comparación (operador de igualdad, método Equals y método GetHashCode).
  + Todas las entidades sobrescribirán el método ToString y retornarán una cadena de texto con los datos del objeto.
  + Al menos una jerarquía de herencia que aproveche el pilar del polimorfismo.
  + Todas las entidades deberán estar correctamente encapsuladas exponiendo sólo sus operaciones y características esenciales, protegiendo el acceso y modificación libre de datos, y ocultando los detalles de la implementación.
  + Al menos una sobrecarga de constructores y una sobrecarga de métodos.
  + Clases, atributos y métodos que no correspondan o trabajen con el estado de una instancia particular deberán ser estáticos.
  + Todos los objetos deberán inicializar su estado con los mínimos valores necesarios para que no exista lugar a fallos en el uso del objeto, no debiendo permitir que se instancien de otra forma.
  + Declaración y uso de al menos un enumerado.
  + Declaración y uso de al menos un indexador.
  + Declaración y uso de al menos dos tipos de colecciones genéricas distintas.
  + Uso justificado de al menos una clase abstracta y una clase sellada.

**❗❗ Si no se cumplen TODAS las condiciones mínimas de aprobación, no se continuará con la corrección y la nota será un dos (desaprobado).**

Además de cumplir con las condiciones mínimas, se deberá alcanzar el puntaje indicado en la sección [calificación](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#calificaci%C3%B3n).

**Condiciones mínimas para promocionar**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#condiciones-m%C3%ADnimas-para-promocionar)

Para promocionar se deberá cumplir todas las condiciones mínimas de aprobación y proponer una nueva funcionalidad en base al contexto del negocio (que no sean las propuestas para los recuperatorios).

El nuevo feature deberá ser agregado a la aplicación siguiendo todos los criterios de calidad y buenas prácticas antes nombrados.

Justificar el valor agregado de su elección en la sección *"Propuesta de valor agregado"* del archivo README.md.

**❗❗ Si no se cumplen TODAS las condiciones mínimas de promoción, se corregirá el parcial pero la nota no podrá superar el cuatro (aprobado sin promoción).**

Una vez cumplan con las condiciones, se los irá llamando **en cualquier momento de la cursada** para defender el proyecto de forma oral y contar cómo lo resolvieron. Para esto deberán contar con cámara y micrófono, pudiendo conectarse desde un celular si fuera necesario. La exposición será grabada y se deberá presentar libreta universaria o documento de identidad.

El resultado de la exposición oral **puede anular la promoción** si detectamos que no conocen en profundidad el proyecto o no entienden lo que hicieron. Por el contrario, una muy buena exposición sumará un punto a la calificación. Habrá casos donde no se sume el punto pero se mantenga la promoción.

Además de cumplir con las condiciones mínimas, se deberá alcanzar el puntaje indicado en la sección [calificación](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#calificaci%C3%B3n).

**Calificación**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#calificaci%C3%B3n)

**Una vez que se hayan superado las condiciones de aprobación**, se calificará por tema evaluando un uso correcto y bien justificado de cada uno de los temas vistos entre la clase 01 y la 10.

| **Tema** | **Peso** |
| --- | --- |
| Reglas de estilo, buenas prácticas y calidad de documentación. | 1 punto |
| Algoritmos y resolución de problemas. Reutilización de código. Elección de tipos de datos. Trabajo con cadenas de texto, fechas, etc. | 2 puntos |
| Uso correcto y justificado de miembros estáticos. | 1 punto |
| Abstracción. Diseño de clases y objetos (diagrama de clases, atributos, métodos, constructores). | 2 puntos |
| Uso correcto y justificado de sobrecargas de métodos, operadores y constructores. | 1 punto |
| Uso correcto y justificado de colecciones. | 1 punto |
| Creatividad en el diseño de formularios. Uso correcto de controles y eventos. Buena experiencia de usuario. | 2 puntos |
| Encapsulamiento y uso correcto y justificado de modificadores de acceso. | 2 puntos |
| Uso correcto y justificado de propiedades, enumerados e indexadores. | 1 puntos |
| Uso correcto y justificado de jerarquías de herencia, clases abstractas y selladas. | 2 puntos |
| Uso correcto y justificado de polimorfismo. | 2 puntos |
| Solución robusta, con validaciones y control de errores. | 1 punto |
| Investigación, instalación y uso de una biblioteca externa (paquete nuget) que resuelva o ayude a resolver alguna necesidad de la implementación. | 1 punto |
| Defensa oral. | 1 punto |
| **Total** | **20 puntos** |

| **Puntos** | **Nota** |
| --- | --- |
| >= 10 && < 12 | 4 |
| >= 12 && < 14 | 6 |
| >= 14 && < 16 | 7 |
| >= 16 && < 18 | 8 |
| >= 18 && < 20 | 9 |
| 20 | 10 |

**Entrega**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#entrega)

* Con su cuenta de GitHub deberá realizar un *fork* al siguiente repositorio: <https://github.com/mauricioCerizza/2021C2_UTNFra_LaboII_2doD_PP>.
  + Trabajar sobre el mismo.
  + Las entregas se deberán realizar en el branch Master.
* Cada cambio y avance deberá ser registrado con un *commit* puntual acompañado de un mensaje descriptivo y subido inmediatamente al repositorio remoto (*push*).
* La solución deberá ser nombrada como: *Apellido.Nombre.PrimerParcial*.
* No realizar actualizaciones sobre el branch Master entre la fecha de entrega y la corrección. Si desean seguir trabajando pueden hacerlo sobre otra ramificación.
* Registrar **cada** entrega a través del Google Form indicado a continuación. **No se corregirán repositorios que no estén registrados** en la correspondiente instancia de entrega.

| **Instancia** | **Fecha** | **Formulario de registro** |
| --- | --- | --- |
| Revisión (opcional) | Lunes 27/09/2021 10:00 AM | [Registrá tu repositorio para la instancia de revisión](https://forms.gle/crZLsvpP26KyaRHs9) |
| Primera entrega | Lunes 11/10/2021 10:00 AM | [Registrá tu repositorio para la instancia de primera entrega](https://forms.gle/aEdKgSB7fuZ3ooHQ7) |
| Primer recuperatorio | Lunes 25/10/2021 10:00 AM | [Registrá tu repositorio para la primer instancia de recuperatorio](https://forms.gle/vAE8fP8SowCvkruh6) |
| Segundo recuperatorio | Lunes 29/11/2021 10:00 AM | [Registrá tu repositorio para la segunda instancia de recuperatorio](https://forms.gle/4SxYXripefuU4gcNA) |

Los trabajos serán descargados en la fecha y hora indicada, siendo corregidos a partir de ese momento según criterio docente. Existirá una instancia de revisión **opcional**, donde se podrá entregar el trabajo (completo o incompleto) y recibir una corrección preliminar que no afectará la nota final.

También se podrán utilizar los espacios de consulta en clase para evacuar cualquier duda o inconveniente que no hayan podido solucionar. **No se responderán mensajes privados por Slack fuera del horario de clase.**

En cada recuperatorio se deberá entregar TODO lo requerido en la instancia anterior completo y con las correcciones indicadas, y además se agregarán [nuevas consignas y requerimientos adicionales](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#primer-recuperatorio).

El segundo recuperatorio reemplazará a la instancia de recuperación en **primera** fecha de final, donde deberán asistir sólo quienes estén en condiciones de promoción y aún no hayan defendido el trabajo.

El parcial **debe realizarse de forma individual**, sin colaboración, ni corrección, ni guía, ni nada similar de ningún compañero.

**ADVERTENCIA**

La sospecha o detección de copias conducirá a la desaprobación de ambos alumnos, quedando sin posibilidad de promoción quien tenga el commit más reciente sobre el repositorio.

Para los recuperatorios deberán rehacer el trabajo con un diseño e implementación completamente nuevo y la nota máxima será cuatro.

**Recomendaciones**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#recomendaciones)

* Aprovechar los espacios de consulta de cada clase.
* Si se traban con algo sigan con otra cosa, no pierdan demasiado tiempo en un tema puntual.
* Considerar el deadline (fecha de entrega) y el tiempo de trabajo con el que cuentan. Planificar y priorizar su trabajo. No dejar todo para último momento.
* Foco en el objetivo. Priorizar la entrega completa del trabajo y dejar para el final temas menos importantes (como pueden ser retoques al estilo visual).

**Primer recuperatorio**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#primer-recuperatorio)

Para aprobar el primer recuperatorio se deberán cumplir con TODAS las consignas y requisitos de la primera entrega, arreglar todo lo que se haya marcado en las correcciones y sumar las siguientes funcionalidades:

* Agregar una pantalla de login, si no se ingresa un usuario y contraseña válidos no se podrá acceder a las otras pantallas. Debe existir la posibilidad de desloguearse y loguearse con otro usuario sin cerrar la aplicación.
* Manejar los siguientes perfiles de usuario: **operador** y **técnico**. El perfil deberá figurar en la barra de información junto al nombre del usuario.

**Operador (a.k.a "el pibe que atiende el ciber")**

* Además de las funciones que ya tenía (asignar y deasignar), deberá poder asociar problemas técnicos a una máquina o teléfono.
  + Deberá existir una lista de opciones pre-cargadas y la posibilidad de ingresar una razón que no se encuentre en la lista.
  + Ejemplos: *"No anda el mouse."*, *"Se cortó el cable."*, *"No tiene cámara."*.
  + Al cargar un problema deberá asignarle una prioridad entre las siguientes: Alta, Media o Baja.
  + Las máquinas o cabinas que tengan un problema asociado y no arreglado deberán figurar en estado *"Inhabilitado"*.

Usuario: Oper Contraseña: PasáPorLaNueve

Copy

**Técnico (a.k.a "el que estudió programación pero lo llamaron para arreglar impresoras")**

* Podrá visualizar solamente los problemas técnicos pendientes ordenados por prioridad de los más prioritarios a los menos prioritarios, como segundo criterio ordenar alfabéticamente de forma ascendente.
* Sólo él puede marcar los problemas como arreglados.

Usuario: CasiHackerContraseña: TeFormateoElDisco

Copy

**Segundo recuperatorio**[**#**](https://codeutnfra.github.io/programacion_2_laboratorio_2_apuntes/docs/evaluaciones/parciales/2d-primer-parcial#segundo-recuperatorio)

Para aprobar el segundo recuperatorio se deberán cumplir con TODAS las consignas y requisitos de la primera entrega y del primer recuperatorio, arreglar todo lo que se haya marcado en las correcciones y sumar las siguientes funcionalidades:

Agregar el perfil **administrador (a.k.a "el dueño")** que tendrá todas las funciones disponibles (incluyendo las del técnico) y se le sumarán las siguientes exclusivas de este perfil:

* Poder cambiar los costos/tarifas de los servicios.
* Poder acceder a un **historial de acciones** realizadas por los distintos usuarios partiendo de la última hasta la primera.
  + Deberá contener la **fecha y hora** exacta en que sucedió, el **nombre** del usuario, su **perfil**, y la **descripción** de la acción.
* Será el único que pueda ver las estadísticas históricas. El operador ya no podrá ver esta información.

Usuario: Admin Contraseña: Admin

Copy

Agregar el perfil de **cliente** a la aplicación, que no requerirá usuario y contraseña para ingresar. Se montará una computadora en la entrada del local donde los clientes podrán pre-cargar sus datos en la aplicación y auto-asignarse a la cola de espera.

Entre los datos a cargar se encontrarán:

* Nombre
* Apellido
* DNI
* Fecha de nacimiento
* Mail
* Elegir cabina o máquina.
  + Si elige máquina, deberá seleccionar uno o varios programas, juegos o periféricos a utilizar. También ingresar la cantidad de fracciones de tiempo a alquilar o modo *"Libre"*.
  + Si elige cabina, deberá ingresar el teléfono al que llamará (mismo criterio que en las entregas anteriores).

Integrar esta nueva funcionalidad con las existentes. Los datos cargados por el cliente ya no deberán ser ingresados por el operador.

*Last updated on****9/14/2021****by****mauriciocerizza***