

Informe final taller 3

de

programación avanzada

2025

Geancarlo Bisquett

21.378.865-5

[geancarlo.bisquett@alumnos.ucn.cl](mailto:geancarlo.bisquett@alumnos.ucn.cl)

Profesor:

• Tomás Reimann ([tomas.reimann@ce.ucn.cl](mailto:tomas.reimann@ce.ucn.cl))

Ayudantes:

• Rodrigo Aguilera (rodrigo.aguilera01@alumnos.ucn.cl)

• Diego Cortés ([diego.cortes07@alumnos.ucn.cl](mailto:diego.cortes07@alumnos.ucn.cl))

Introducción:

Actualmente, el proceso de creación y postulación a ofertas académicas dentro de la Universidad Católica del Norte (UCN) se realiza de forma manual, lo que genera múltiples problemas de gestión como falta de trazabilidad, duplicación de información, demoras en los procesos y dificultad para acceder a los registros. Este escenario afecta tanto a docentes como a estudiantes, quienes no cuentan con una plataforma unificada que centralice la información de ayudantías, capstone y prácticas pre-profesionales.

Para abordar esta problemática, se propone el desarrollo de un sistema digital que automatice la creación, gestión y visualización de ofertas académicas. Este sistema permite:

* El registro y autenticación de usuarios mediante Rut y contraseña.
* La creación de ofertas académicas según su tipo (Ayudantía, Capstone o Práctica), validando los datos obligatorios según cada caso.
* La visualización y búsqueda de ofertas por código o título.
* La edición y eliminación de ofertas existentes.
* La consulta y edición del perfil del usuario autenticado.
* El manejo centralizado de todas las ofertas mediante una interfaz y una clase concreta que actúa como sistema.
* El código está estructurado de forma que cada operación del sistema está validada mediante contratos, definidos en la interfaz SistemaOferta. Por ejemplo, la operación ingresarOferta() valida que los campos requeridos estén completos y que el tipo de oferta sea válido, antes de registrarla dentro del contenedor de ofertas.
* Para manejar las distintas ofertas, se emplea herencia a través de la clase OfertaAcademica, lo que permite tratar todas las ofertas de forma uniforme desde el sistema, pero a la vez diferenciarlas internamente según su tipo (ayudantía, capstone, práctica).
* Además, se utiliza un contenedor específico (contenedor oferta) que implementa un arreglo con control de tamaño y posición, lo cual permite operar sin depender de estructuras dinámicas externas. Cada búsqueda, edición o eliminación se realiza validando primero si la oferta existe en el sistema.
* Esta solución mejora significativamente la organización del proceso, evita duplicidad de datos, permite buscar ofertas fácilmente y asegura la integridad del sistema mediante validaciones en cada operación.

2. Horas dedicadas

| Nombre del Estudiante | Diseño (hrs) | Codificación (hrs) | Informe (hrs) | Total (hrs) |
| --- | --- | --- | --- | --- |
| [Geancarlo Bisquett | 3 | 9 | 2 | 14 |
|  |  |  |  |  |

3.Contratos:

Se mantienen los mismos contratos.

|  |  |
| --- | --- |
| Operación: | registrarUsuario(Rut:cadena de texto, Nombre:cadena de texto, Email:cadena de texto, Contraseña:cadena de texto) |
| Descripción: | Permite registrar un nuevo usuario en el sistema, almacenando su nombre, Rut, correo y contraseña. El Rut será el identificador principal. Esta operación solo está disponible si el usuario aún no está registrado. |
| Pre-Condiciones | El nombre debe tener entre 3 y 255 caracteres, el correo debe tener  formato válido y ser único, la contraseña debe tener entre 6 y 20  caracteres.  El Rut debe:  Tener 8 dígitos numéricos más un dígito verificador correcto.  No estar repetido en el sistema.  No contener espacios ni caracteres inválidos. |
| Post-Condiciones | El usuario queda registrado correctamente en el sistema y puede iniciar sesión correctamente con su Rut y contraseña |

|  |  |
| --- | --- |
| Operación: | iniciarSesion(Rut:cadena de texto, Contraseña:cadena de texto) |
| Descripción: | Permite a un usuario autenticarse en el sistema ingresando su RUT y contraseña. Solo los usuarios registrados pueden iniciar sesión |
| Pre-Condiciones | El Rut debe estar registrado en el sistema, la contraseña debe coincidir con la registrada, el RUT y la contraseña no deben estar vacíos. |
| Post-Condiciones | El usuario queda autenticado y puede acceder a las funcionalidades restringidas. Se mantiene activa la sesión hasta que el usuario la cierre. |

|  |  |
| --- | --- |
| Operación: | cerrarSesion() |
| Descripción: | Permite cerrar la sesión del usuario actual y retornar al menú de inicio. |
| Pre-Condiciones | Debe haber un usuario con sesión iniciada. |
| Post-Condiciones | Se termina la sesión actual y se vuelve al estado inicial del sistema (sin usuario activo). |

|  |  |
| --- | --- |
| Operación: | ingresarOferta(tipoDeOferta:número entero, codigoUnico:número entero, titulo:cadena de texto, descripción:cadena de texto, duracionEnDias:número entero) |
| Descripción: | permite registrar una nueva oferta académica en el sistema. El usuario debe ingresar un conjunto de datos comunes a todas las ofertas (como código, título, descripción y duración), además de información adicional específica según el tipo de oferta seleccionada: ayudantía, capstone o práctica pre-profesional. El sistema valida que todos los campos estén correctamente ingresados y que el código sea único antes de guardar la oferta. |
| Pre-Condiciones | Para que la operación se cumpla el código, título, descripción, ni ninguna información solicitada debe ser nulos ni vacíos. El código tiene que ser único. La duración de días debe ser mayor a 0 y el tipo de oferta debe ser uno de los siguientes: “Ayudantía”, “Capstone”, “Práctica”. Además, dependiendo del tipo de oferta, se debe cumplir los siguientes requisitos:  Ayudantia: El nombre de la asignatura y el rol del ayudante no pueden ser vacíos o nulos, las horas semanales deben ser mayores a 0 y el promedio mínimo debe estar entre 1.0 y 7.0.  Capstone: Fecha de inicio, nombre de la empresa, nombre del guía y el tipo de proyecto no pueden ser vacíos, la duración (en meses) debe ser mayor a 0, se debe tener al menos una carrera, y la mínima cantidad de estudiantes debe ser mayor a 0.  Practica: Fecha de inicio, nombre de la empresa, nombre del guía no pueden ser vacíos ni nulos, si tiene remuneración debe estar definido en sí o no (verdadero o falso). |
| Post-Condiciones | La nueva oferta queda registrada en el sistema, los datos ingresados se almacenan correctamente según el tipo de oferta, el sistema asegura que el código es único y valido. |

|  |  |
| --- | --- |
| Operación: | ingresarOferta(tipoDeOferta:número entero, codigoUnico:número entero, titulo:cadena de texto, descripción:cadena de texto, duracionEnDias:número entero) |
| Descripción: | permite registrar una nueva oferta académica en el sistema. El usuario debe ingresar un conjunto de datos comunes a todas las ofertas (como código, título, descripción y duración), además de información adicional específica según el tipo de oferta seleccionada: ayudantía, capstone o práctica pre-profesional. El sistema valida que todos los campos estén correctamente ingresados y que el código sea único antes de guardar la oferta. |
| Pre-Condiciones | Para que la operación se cumpla el código, título, descripción, ni ninguna información solicitada debe ser nulos ni vacíos. El código tiene que ser único. La duración de días debe ser mayor a 0 y el tipo de oferta debe ser uno de los siguientes: “Ayudantía”, “Capstone”, “Práctica”. Además, dependiendo del tipo de oferta, se debe cumplir los siguientes requisitos:  Ayudantia: El nombre de la asignatura y el rol del ayudante no pueden ser vacíos o nulos, las horas semanales deben ser mayores a 0 y el promedio mínimo debe estar entre 1.0 y 7.0.  Capstone: Fecha de inicio, nombre de la empresa, nombre del guía y el tipo de proyecto no pueden ser vacíos, la duración (en meses) debe ser mayor a 0, se debe tener al menos una carrera, y la mínima cantidad de estudiantes debe ser mayor a 0.  Practica: Fecha de inicio, nombre de la empresa, nombre del guía no pueden ser vacíos ni nulos, si tiene remuneración debe estar definido en sí o no (verdadero o falso). |
| Post-Condiciones | La nueva oferta queda registrada en el sistema, los datos ingresados se almacenan correctamente según el tipo de oferta, el sistema asegura que el código es único y valido. |

|  |  |
| --- | --- |
| Operación: | buscarOferta(codigoUnico:número entero, título: cadena de texto) |
| Descripción: | Permite buscar una oferta específica por su código único o título. Devuelve toda la información relacionada a esa oferta. |
| Pre-Condiciones | Debe existir al menos una oferta registrada. El código o título ingresado no debe estar vacío. |
| Post-Condiciones | Si se encuentra la oferta, se muestra su información completa. Si no se encuentra, se notifica al usuario. |

|  |  |
| --- | --- |
| Operación: | verOfertas() |
| Descripción: | Muestra en pantalla todas las ofertas registradas en el sistema con sus datos principales de forma clara y ordenada. |
| Pre-Condiciones | Debe haber al menos una oferta registrada. |
| Post-Condiciones | Se imprime un listado con el código, título, tipo y resumen breve de cada oferta. |

|  |  |
| --- | --- |
| Operación: | editarOferta(codigoUnico:numero entero) |
| Descripción: | Permite editar una oferta existente a través de su código único, reemplazando todos los datos anteriores por nuevos valores validados. |
| Pre-Condiciones | Usuario debe estar autenticado, el código debe existir en el sistema y nuevos datos deben cumplir todas las validaciones de ingresarOferta. |
| Post-Condiciones | La oferta se actualiza con los nuevos datos proporcionados. |

|  |  |
| --- | --- |
| Operación: | eliminarOferta(codigoUnico:número entero) |
| Descripción: | Elimina una oferta del sistema utilizando su código único. Antes de realizar la acción, se debe confirmar la eliminación. |
| Pre-Condiciones | El usuario debe estar autenticado, el código debe existir en el sistema, se debe haber conseguido confirmación del usuario. |
| Post-Condiciones | La oferta se eliminara del registro, no será posible acceder a esa oferta desde búsquedas siguientes. |

|  |  |
| --- | --- |
| Operación: | verPerfil() |
| Descripción: | Permite al usuario autenticado visualizar su nombre, correo y RUT registrados en el sistema (sin mostrar contraseña). |
| Pre-Condiciones | Debe haber un usuario autenticado. |
| Post-Condiciones | Se imprime por consola la información del perfil del usuario actual (excluyendo la contraseña). |

|  |  |
| --- | --- |
| Operación: | editarPerfil() |
| Descripción: | Permite al usuario cambiar su nombre y correo electrónico. Los nuevos datos deben ser validados antes de su registro. |
| Pre-Condiciones | El usuario debe estar autenticado, el nuevo nombre a ingresar y correo debe ser válidos, el nuevo correo no debe estar repetido en el sistema. |
| Post-Condiciones | Los datos del perfil del usuario son actualizados correctamente. |

|  |  |
| --- | --- |
| Operación: | cerrarSesion() |
| Descripción: | Finaliza la sesión del usuario actualmente autenticado y retorna al menú inicial del sistema. A partir de este momento, el usuario deberá volver a iniciar sesión para acceder a las funcionalidades restringidas. |
| Pre-Condiciones | Debe haber un usuario con sesión iniciada. |
| Post-Condiciones | La sesión del usuario actual se cierra, el sistema queda en estado inicial, esperando un nuevo inicio de sesión o registro. |

4.Diagrama de clases

