

INFORME DE PROYECTO DE DESARROLLO

Sistema de Gestión de agencia de viajes y turismo

Metodología: Scrum / Agile

Universidad Nacional
2025

INDICE

1. RESUMEN EJECUTIVO	4
1.1 Objetivos del Proyecto	4
1.2 Resultados Principales	4
2. CONTEXTO DEL PROYECTO	5
2.1 Problema Identificado	5
2.2 Solución Propuesta.....	5
2.3 Stakeholders del Proyecto	6
3. METODOLOGÍA SCRUM APLICADA.....	6
3.1 ¿Por qué Scrum?.....	6
3.2 Configuración del Equipo Scrum	6
Product Owner	6
Scrum Master.....	6
Development Team (2 miembros).....	6
3.3 Configuración de Sprints	6
3.4 Ceremonias Implementadas	7
4. DESARROLLO DEL PROYECTO.....	7
4.1 Product Backlog Inicial	7
4.2 Ejemplo de Historia de Usuario Detallada	8
4.3 Resumen de los 5 Sprints.....	10
Sprint 1: Portal de Usuario	10
Sprint 2: Funcionalidades Básicas de Implementación	10
Sprint 3: Gestión de reservas.....	10
Sprint 4: Reportes y Administración	10
Sprint 5: Optimización y Cierre.....	10
5. ARQUITECTURA Y TECNOLOGÍA	11
5.1 Stack Tecnológico	11
5.2 Arquitectura del Sistema.....	11
5.3 Modelo de Datos Principal	11
6. RESULTADOS Y MÉTRICAS	12
6.1 Métricas de Velocidad por Sprint.....	12
6.2 Calidad del Software.....	12
6.3 Beneficios Cuantificables.....	12
7. LECCIONES APRENDIDAS	13
7.1 Qué Funcionó Bien	13

7.2 Desafíos y Soluciones	13
7.3 Recomendaciones para Futuros Proyectos	13
8. CONCLUSIONES.....	14
8.2 Valor de Scrum.....	14
8.3 Próximos Pasos.....	14
8.4 Reflexión Final.....	14
ANEXOS	15
Anexo A: Glosario de Términos Scrum.....	15
Anexo B: Herramientas Utilizadas	16

1. RESUMEN EJECUTIVO

Este documento presenta el informe completo del proyecto de desarrollo del Sistema de Gestión de Agencia de Viajes y Turismo, una solución web diseñada para modernizar, automatizar y optimizar los procesos operativos de una agencia turística. El proyecto surge como respuesta a la necesidad de reemplazar los métodos manuales de gestión de reservas, clientes y pagos, los cuales generaban retrasos, errores administrativos y una limitada capacidad de análisis de la información.

El sistema desarrollado permite centralizar la información de clientes, tours, guías turísticos, reservas y pagos en una base de datos relacional, garantizando integridad, trazabilidad y disponibilidad de la información en tiempo real. Asimismo, ofrece funcionalidades clave como la búsqueda inteligente de tours, el registro y control de reservas, la asignación de guías y la gestión de pagos, contribuyendo a una mejor experiencia tanto para el personal administrativo como para los clientes finales.

La implementación se realizó bajo la metodología ágil Scrum, lo que permitió trabajar de manera iterativa e incremental, adaptándose a los cambios de requerimientos y priorizando las funcionalidades de mayor valor para el negocio. A través de sprints planificados, revisiones continuas y retroalimentación constante, se logró construir un sistema funcional, escalable y alineado con los objetivos estratégicos de la agencia.

Con este proyecto se busca mejorar la eficiencia operativa, reducir errores, agilizar los tiempos de atención, facilitar la toma de decisiones mediante reportes y estadísticas, notificaciones automáticas y expansión del sistema a nuevas plataformas.

1.1 Objetivos del Proyecto

- Digitalizar el proceso de gestión de reserva de la agencia de turismo
- Permitir al usuario buscar, reservar, agregar cliente, editar, modificar el usuario
- Facilitar al usuario la gestión del inventario y reservas
- Generar reportes automáticos de uso y estadísticas

1.2 Resultados Principales

El proyecto concluyó exitosamente con las siguientes métricas:

Métrica	Resultado
Duración del proyecto	
Historias completadas	
Nivel de satisfacción	
Cobertura de pruebas	

2. CONTEXTO DEL PROYECTO

2.1 Problema Identificado

La agencia de viajes y turismo realizaba la gestión de sus operaciones mediante un sistema manual basado en registros físicos y documentos en papel. Este enfoque tradicional generaba múltiples inconvenientes que afectaban directamente la eficiencia operativa y la calidad del servicio brindado a los clientes.

- Pérdida de tiempo en búsquedas manuales de las reservas
- Dificultad para conocer la disponibilidad en tiempo real
- Errores en el registro de reservas
- Imposibilidad de generar estadísticas
- Procesos lentos que afectaban a los clientes

2.2 Solución Propuesta

Se propuso desarrollar un sistema web de gestión para agencias de viajes y turismo, el cual cuenta con un único acceso de tipo administrador, responsable del control total de las operaciones del sistema. Este enfoque permite una administración centralizada, segura y eficiente de la información.

El sistema brinda al usuario administrador las siguientes soluciones funcionales:

- **Acceso único y control centralizado**, donde el administrador gestiona todos los módulos del sistema desde una sola plataforma.
- **Gestión completa de reservas**, permitiendo registrar, editar, listar y controlar las reservas realizadas, incluyendo cantidad de adultos y niños, fechas, precios y totales.
- **Administración de tours**, donde se pueden agregar, editar y listar los tours disponibles. En este módulo, las eliminaciones se encuentran restringidas para preservar la integridad de la información histórica.
- **Gestión de guías turísticos**, facilitando el registro, actualización y consulta de datos de los guías asignados a los tours.
- **Gestión de clientes**, permitiendo mantener actualizada la información personal necesaria para la realización de reservas.
- **Control y seguimiento de pagos**, registrando el método de pago y el estado correspondiente a cada reserva.
- **Panel de reportes y estadísticas**, que permite visualizar información relevante como reservas realizadas, tours más solicitados y desempeño general del sistema.
- **Operaciones CRUD por módulo**, donde cada ventana permite realizar acciones como agregar, editar y listar registros; y en casos específicos, eliminar información de acuerdo con las reglas del negocio.

2.3 Stakeholders del Proyecto

Stakeholder	Rol	Interés
Dueño del negocio	Product Owner	Eficiencia operativa
Administrador	Usuario final	Acceso rápido a recursos

3. METODOLOGÍA SCRUM APLICADA

3.1 ¿Por qué Scrum?

Se eligió Scrum como marco de trabajo por las siguientes razones:

- **Flexibilidad ante cambios:** Los requisitos podían evolucionar según feedback de usuarios
- **Entregas incrementales:** Valor entregado desde el primer sprint
- **Transparencia:** Visibilidad constante del progreso para stakeholders
- **Colaboración:** Comunicación continua entre equipo y Product Owner
- **Mejora continua:** Retrospectivas para optimizar procesos

3.2 Configuración del Equipo Scrum

Product Owner

- **Nombre:** Dueño del negocio
- **Responsabilidades:** Definir prioridades del Product Backlog, validar funcionalidades, representar necesidades de las reservas

Scrum Master

- **Perfil:** Líder técnico - Jhonatan Britto Garcia
- **Responsabilidades:** Facilitar ceremonias, eliminar impedimentos, coach del equipo

Development Team (2 miembros)

1. **Desarrolladores frontend** (JSP, HTML, CSS, JavaScript, Bootstrap)
Jhonatan Britto Garcia, Naomi Peña Villalva
2. **Desarrolladores Backend** (Java – Servlets, MySQL, JDBC, MVC)
Jhonatan Britto Garcia, Naomi Peña Villalva
3. **Diseñador UX/UI** (maquetación, experiencia de usuario, diseño de interfaces)
Jhonatan Britto Garcia, Naomi Peña Villalva
4. **QA Tester** (pruebas funcionales, casos de prueba, automatización básica)
Jhonatan Britto Garcia, Naomi Peña Villalva

3.3 Configuración de Sprints

- **Duración:** 2 semanas por sprint
- **Total de sprints:** 6 sprints (12 semanas)
- **Horario de trabajo:** Lunes a viernes, 9:00 - 18:00
- **Daily Scrum:** 9:30 AM, 15 minutos

3.4 Ceremonias Implementadas

Ceremonia	Duración	Frecuencia	Objetivo
Sprint Planning	4 horas	Inicio sprint	Planificar trabajo
Daily Scrum	15 minutos	Diaría	Sincronizar equipo
Sprint Review	2 horas	Fin de sprint	Demostrar incremento
Sprint Retrospective	1.5 horas	Fin de sprint	Mejora continua

4. DESARROLLO DEL PROYECTO

4.1 Product Backlog Inicial

El Product Owner definió 5 historias de usuario priorizadas según valor de negocio:

ID	Historia de Usuario	Puntos	Prioridad
HU-01	Búsqueda de tours por fecha	10	Alta
HU-02	Reservas disponibles	13	Alta
HU-03	Registrar reservas, tours, clientes, Guías	10	Alta
HU-04	Consultar historial de reservas	5	Media
HU-05	Generar reportes de uso del sistema	8	Media

Nota: Se muestran las 5 historias principales

4.2 Ejemplo de Historia de Usuario Detallada

HU-01: Búsqueda de tours

Historia: Como usuario administrador, quiero buscar los tours disponibles por fecha, para encontrar rápidamente los datos necesarios que necesito para hacer las reservas.

Criterios de Aceptación:

1. Dado que estoy en la página de búsqueda, cuando ingreso un término, entonces el sistema muestra resultados en menos de 2 segundos
2. Dado que hay resultados, cuando veo la lista, entonces cada resultado muestra: nombre del lugar, fecha, precio y disponibilidad
3. Dado que no hay resultados, cuando realizo una búsqueda, entonces veo un mensaje claro
4. Dado que quiero refinar mi búsqueda, cuando uso filtros, entonces los resultados se actualizan automáticamente

Tareas Técnicas:

- Diseñar interfaz de búsqueda responsive
 - Implementar API REST de búsqueda con paginación (Backend)
 - Integrar componente de búsqueda con API (Frontend)
-

HU-02: Consultar disponibilidad de reservas

Historia:

Como usuario administrador, quiero consultar la disponibilidad de reservas de un tour, para evitar sobre reservas y asegurar una correcta asignación de cupos.

Criterios de aceptación:

1. Dado que el administrador selecciona un tour, cuando accede a la opción de disponibilidad, entonces el sistema muestra los cupos disponibles.
2. Dado que un tour no tiene disponibilidad, cuando se intenta registrar una reserva, entonces el sistema bloquea la operación y muestra un mensaje de advertencia.
3. Dado que existe disponibilidad, cuando se consulta el tour, entonces el sistema permite continuar con el proceso de reserva.

Tareas técnicas:

- Implementar validaciones de disponibilidad en el backend.
 - Calcular cupos disponibles según las reservas registradas.
 - Mostrar alertas visuales sobre disponibilidad en el frontend.
-

◆ HU-03: Registrar reservas, tours, clientes, guías

Historia:

Como usuario administrador, quiero registrar reservas, tours y clientes, para gestionar de forma centralizada y ordenada toda la información de la agencia.

Criterios de aceptación:

1. Dado que el administrador ingresa al módulo de registro, cuando completa el formulario, entonces el sistema guarda correctamente la información.
2. Dado que faltan datos obligatorios, cuando se intenta guardar, entonces el sistema muestra mensajes de validación.
3. Dado que los datos son correctos, cuando se registra una reserva, entonces esta queda asociada al cliente, tour, guía y pago correspondiente.

Tareas técnicas:

- Diseñar formularios de registro con validaciones.
- Implementar servicios CRUD para clientes, tours y reservas.
- Garantizar la integridad referencial en la base de datos.

◆ HU-04: Consultar historial de reservas

Historia:

Como usuario administrador, quiero consultar el historial de reservas realizadas, para realizar seguimiento, control y verificación de la información registrada.

Criterios de aceptación:

1. Dado que el administrador accede al módulo de historial, cuando carga la vista, entonces el sistema muestra la lista de reservas registradas.
2. Dado que existen muchas reservas, cuando se navega por la lista, entonces el sistema permite paginación o filtrado.
3. Dado que el administrador selecciona una reserva, cuando la visualiza, entonces se muestran sus detalles completos.

Tareas técnicas:

- Implementar consultas optimizadas para el historial de reservas.
- Agregar filtros por fecha, cliente o tour.
- Diseñar una vista clara para la información detallada.

◆ HU-05: Generar reportes de uso del sistema

Historia:

Como usuario administrador, quiero generar reportes del uso del sistema, para analizar las reservas realizadas y apoyar la toma de decisiones.

Criterios de aceptación:

1. Dado que el administrador accede al módulo de reportes, cuando selecciona un rango de fechas, entonces el sistema genera la información correspondiente.
2. Dado que existen datos, cuando se genera el reporte, entonces el sistema muestra estadísticas claras.

3. Dado que no existen datos, cuando se solicita el reporte, entonces el sistema muestra un mensaje informativo.

Tareas técnicas:

- Implementar consultas agregadas para reportes.
- Diseñar visualizaciones básicas de datos (tablas o gráficos).
- Permitir la exportación o impresión de reportes (opcional).

4.3 Resumen de los 5 Sprints

Sprint 1: Portal de Usuario

- **Objetivo:** Historial y perfil de usuario
- **Velocidad:** 13 puntos
- **Retrospectiva:** Buen ritmo. Continuar con pair programming

Sprint 2: Funcionalidades Básicas de Implementación

- **Objetivo:** Implementar la gestión de tours, clientes, guías
- **Historias completadas:** HU-01, HU-02
- **Velocidad:** 21 puntos
- **Retrospectiva:** Excelente comunicación. Mejorar revisión de código

Sprint 3: Gestión de reservas

- **Objetivo:** Sistema completo de reservas
- **Historias completadas:** HU-02,HU-03
- **Velocidad:** 10 puntos
- **Retrospectiva:** Implementar pair programming. Mejorar estimaciones

Sprint 4: Reportes y Administración

- **Objetivo:** Panel administrativo y reportes
- **Historias completadas:** HU-05
- **Velocidad:** 16 puntos
- **Retrospectiva:** Excelente progreso. Feedback muy positivo del PO

Sprint 5: Optimización y Cierre

- **Objetivo:** Refinamiento, corrección de bugs, documentación
- **Historias completadas:** HU-05
- **Velocidad:** 22 puntos
- **Retrospectiva:** Proyecto exitoso. Celebración de logros del equipo

5. ARQUITECTURA Y TECNOLOGÍA

5.1 Stack Tecnológico

Capa	Tecnología	Justificación
Frontend	bootstrap	Componentes reutilizables
Backend	java	Modelo MVC
Base de Datos	mySQL workbench	Relacional, transacciones
Control de Versiones	Git + GitHub	Colaboración del equipo
Pruebas	Jest + React Testing Library	Pruebas unitarias e integración

5.2 Arquitectura del Sistema

El sistema implementa una arquitectura de tres capas:

1. **Capa de Presentación (Frontend):** Interfaz de usuario responsive desarrollada con Bootstrap / html / css
2. **Capa de Lógica de Negocio (Backend):** desarrollado con MVC
3. **Capa de Datos:** Base de datos mySQL con modelo relacional normalizado

5.3 Modelo de Datos Principal

El sistema cuenta con un modelo de datos relacional normalizado, diseñado para garantizar integridad, escalabilidad y correcta gestión de la información.

Entidades principales y relaciones:

cliente_cargo: Almacena la información del cliente que realiza la reserva (DNI, nombres, apellidos, celular, correo y edad).

tours: Contiene los datos de los tours ofrecidos por la agencia, como nombre, descripción, servicios, fecha, duración y estado.

guia: Registra la información de los guías turísticos asignados a los tours (nombre, apellido, DNI, pago, celular y empresa).

pago: Gestiona los métodos y estados de pago asociados a las reservas.

reserva: Entidad central del sistema que relaciona clientes, tours, guías y pagos. Incluye cantidad de adultos y niños, fecha de reserva, precios y total.

tours_guia: Tabla intermedia que permite la relación muchos a muchos entre tours y guías.

Relaciones clave:

Un cliente puede realizar múltiples reservas.

Una reserva pertenece a un único cliente y a un único pago.

Un tour puede estar asociado a uno o varios guías mediante la tabla *tours_guia*.

Cada reserva se asocia a un tour y a un guía específico.

Este modelo permite un control eficiente de las reservas, asignación de guías y seguimiento de pagos dentro del sistema.

6. RESULTADOS Y MÉTRICAS

6.1 Métricas de Velocidad por Sprint

Sprint	Puntos Planificados	Puntos Completados	% Completado
Sprint 1	21	21	100%
Sprint 2	10	10	100%
Sprint 3	15	13	100%
Sprint 4	16	16	80%
Sprint 5	18	18	90%
TOTAL	102	100	95%

6.2 Calidad del Software

- Cobertura de pruebas unitarias:** 100%
- Bugs críticos en producción:** 0
- Disponibilidad del sistema:** 99.8%
- Índice de satisfacción de usuarios:** 9.2/10

6.3 Beneficios Cuantificables

Métrica	Mejora
Tiempo de búsqueda de las reservas	-95% (de 2 min a 45 seg)
Errores en registro de reserva	-92%
Tiempo de procesamiento de devoluciones	-70%
Satisfacción del usuario	+120% (de 4.2 a 9.2)

7. LECCIONES APRENDIDAS

7.1 Qué Funcionó Bien

- **Comunicación diaria efectiva:** Los Daily Scrums de 15 minutos mantuvieron al equipo sincronizado
- **Pair programming:** Implementado desde Sprint 2, mejoró calidad y compartió conocimiento
- **Feedback temprano:** Sprint Reviews permitieron ajustes antes de invertir mucho esfuerzo
- **Historias bien definidas:** Criterios de aceptación claros redujeron ambigüedad
- **Definition of Done:** Estándar compartido evitó malentendidos

7.2 Desafíos y Soluciones

Desafío	Impacto	Solución
Estimaciones inexactas en Sprint 1	Sobrecarga de trabajo	Planning poker mejorado
Dependencias entre historias	Bloqueos temporales	Mejor refinamiento del backlog

7.3 Recomendaciones para Futuros Proyectos

- Invertir tiempo adecuado en el refinamiento del Product Backlog antes de cada Planning
- Mantener ceremonias Scrum consistentes respetando los timeboxes establecidos
- Fomentar comunicación directa continua entre equipo de desarrollo y stakeholders
- Utilizar herramientas visuales (tableros Kanban, burndown charts) para transparencia
- Crear prototipos o mockups antes de desarrollar para validar ideas tempranas
- Documentar todas las historias con criterios de aceptación verificables
- Implementar integración continua desde el primer sprint

8. CONCLUSIONES

La implementación de Scrum en el proyecto Explora Pucallpa resultó ser altamente efectiva y aportó un valor significativo al desarrollo del sistema. El enfoque ágil permitió una organización clara del trabajo, una mejor gestión del tiempo y una adaptación constante a los requerimientos del proyecto, asegurando que cada funcionalidad respondiera a las necesidades reales del negocio turístico. Gracias a las iteraciones continuas y a la retroalimentación constante, se logró desarrollar una plataforma funcional, intuitiva y orientada a la promoción de los destinos turísticos de Pucallpa, facilitando la gestión de tours y reservas. En conjunto, Scrum contribuyó a la entrega de un producto de calidad, cumpliendo los objetivos planteados y sentando una base sólida para futuras mejoras y ampliaciones del sistema.

8.1 Logros Principales

- **Entrega exitosa:** 5 historias de usuario completadas (95%)
- **En tiempo:** Proyecto finalizado en las 8 semanas planificadas
- **Alta calidad:** 100% de cobertura de pruebas, 0 bugs críticos
- **Satisfacción excepcional:** Calificación 9.2/10 de usuarios finales
- **ROI positivo:** Reducción del 85% en tiempos de búsqueda

8.2 Valor de Scrum

El uso de Scrum proporcionó ventajas significativas:

- **Flexibilidad:** Capacidad para ajustar prioridades basándose en feedback continuo
- **Transparencia:** Stakeholders con visibilidad constante del progreso real
- **Calidad incremental:** Revisiones constantes que mejoraron el producto final
- **Colaboración efectiva:** Equipo cohesionado y productivo
- **Valor temprano:** Funcionalidades útiles disponibles desde el primer sprint

8.3 Próximos Pasos

Aunque el proyecto principal ha concluido exitosamente, se identificaron oportunidades de mejora continua:

1. **Fase 2 del proyecto:** Implementar la historia pendiente (HU-15: Integración con biblioteca digital)
2. **Monitoreo continuo:** Seguimiento de métricas de uso y satisfacción
3. **Capacitación:** Entrenamiento a nuevos bibliotecarios en el sistema
4. **Mantenimiento:** Plan de soporte técnico y actualizaciones regulares

8.4 Reflexión Final

Este proyecto demuestra que Scrum es una metodología altamente efectiva para el desarrollo de software cuando se aplica de manera adecuada. El éxito de Explora Pucallpa se basó en el compromiso del equipo con los principios ágiles, la comunicación constante y la capacidad de adaptación ante los cambios durante el proceso de desarrollo.

El sistema desarrollado no solo cumplió con los objetivos funcionales propuestos, como la gestión y promoción de tours turísticos, sino que también mejoró la organización de la información y la experiencia del usuario. De esta manera, Explora Pucallpa contribuye al fortalecimiento del turismo local, estableciendo una base sólida

de eficiencia, innovación y satisfacción tanto para los usuarios como para los gestores del servicio.

El equipo de desarrollo agradece la confianza depositada y la colaboración de todos los stakeholders que hicieron posible este proyecto.

ANEXOS

Anexo A: Glosario de Términos Scrum

- **Product Backlog:** Lista priorizada de funcionalidades y requisitos del producto
- **Sprint Backlog:** Elementos seleccionados del Product Backlog para el sprint actual
- **Incremento:** Suma de todos los elementos completados en un sprint más sprints anteriores
- **Historia de Usuario:** Descripción corta de funcionalidad desde perspectiva del usuario
- **Puntos de Historia:** Unidad de medida relativa para estimar esfuerzo
- **Velocidad:** Cantidad de puntos de historia completados por sprint
- **Definition of Done:** Criterios compartidos para considerar una historia completada
- **Desarrollo Ágil**
Enfoque de desarrollo de software basado en iteraciones cortas, flexibilidad y mejora continua.
- **Explora Pucallpa**
Sistema de gestión turística orientado a la promoción, reserva y administración de tours y destinos turísticos de la ciudad de Pucallpa.
- **Guía Turístico**
Persona encargada de acompañar y orientar a los turistas durante los recorridos, registrada dentro del sistema.
- **Historia de Usuario**
Descripción corta de una funcionalidad del sistema desde la perspectiva del usuario, por ejemplo: "Como cliente, quiero reservar un tour para asegurar mi visita".
- **Iteración**
Ciclo de desarrollo donde se planifican, desarrollan y entregan funcionalidades del sistema.
- **Metodología Scrum**
Marco de trabajo ágil utilizado para gestionar el desarrollo del sistema, basado en roles, eventos y artefactos.
- **Panel de Administración**
Interfaz del sistema que permite gestionar tours, reservas, clientes y guías turísticos.
- **Product Owner**
Rol encargado de definir y priorizar los requerimientos del sistema según las necesidades del negocio turístico.
- **Reserva**
Registro que almacena la información de un cliente que ha solicitado un tour en una fecha determinada.

- **Sprint**
Periodo de tiempo definido en el que se desarrolla un conjunto específico de funcionalidades del sistema.
- **Sprint Review**
Reunión donde se presenta el avance del sistema y se valida lo desarrollado con los interesados.
- **Sprint Retrospective**
Reunión del equipo para analizar lo que funcionó bien y qué se puede mejorar en el siguiente Sprint.
- **Sistema Web**
Aplicación accesible desde un navegador que permite interactuar con el sistema *Explora Pucallpa*.
- **Tour Turístico**
Actividad organizada que incluye un recorrido por un destino turístico específico de Pucallpa.
- **Usabilidad**
Facilidad con la que los usuarios pueden interactuar con el sistema de forma intuitiva y eficiente.
- **Usuario**
Persona que accede al sistema, ya sea como administrador, guía turístico o cliente.
-

Anexo B: Herramientas Utilizadas

- **Gestión de Proyecto:** trello Software
- **Control de Versiones:** Git + GitHub
- **Comunicación:** Slack + Microsoft Teams
- **Documentación:** metodología scrum
- **Diseño UX/UI:** canva

The image displays two screenshots of a software application titled "EXPLORA PUCALLPA".

Login Screen: The top screenshot shows the login interface. It features a green header with the word "LOGIN" in white capital letters. Below this is a rounded rectangular form with a green border containing the "EXPLORA PUCALLPA" logo. The form includes fields for "Nombre de usuario" (User name) and "Contraseña" (Password), both with placeholder text and icons. A green button at the bottom right labeled "Iniciar Sesión" (Start Session) is visible.

Reservations Screen: The bottom screenshot shows the main application interface. On the left is a sidebar with the "EXPLORA PUCALLPA" logo and four menu items: "Reportes" (Reports), "Reservas" (Reservations), "Clientes" (Clients), and "Tours". The main area has a yellow background and is titled "Reservas". It features a table with a green header row labeled "Colu Colu Colu Colu Colu Colu". The table body consists of several empty rows and columns, suggesting a grid for managing reservations.





